FINITE ELEMENT SIMULATION METHODS FOR DRY SLIDING WEAR

THESIS

Aaron Chmiel,

AFIT/GAE/ENY/08-M03

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GAE/ENY/08-M03

FINITE ELEMENT SIMULATION METHODS FOR DRY SLIDING WEAR

THESIS

Presented to the Faculty

Department of Aeronautical and Astronautical Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Aeronautical Engineering

Aaron Chmiel, BS

March 2008

AFIT/GAE/ENY/08-M03

FINITE ELEMENT SIMULATION METHODS FOR DRY SLIDING WEAR

Aaron Chmiel, BS

Approved:

/signed/	14 March 2008
Dr. A.N. Palazotto (Chairman)	Date
/signed/	14 March 2008
Dr. W. Baker (Member)	Date
/signed/	14 March 2008
Dr. R. Brockman (Member)	Date

AFIT/GAE/ENY/08-M03

Abstract

The Holloman High Speed Test Track is a rocket sled track for testing at hypersonic velocities. However, there are customers that desire to test at even greater velocities. In order to achieve higher velocities there are several phenomena that must be overcome. One important phenomenon is wear of the shoe that holds the sled on the rail. This research is a look at the feasibility of using finite element analysis to predict the wear of the shoe during a test run down the track. Two methods are investigated, one is a macro-scale, incremental method utilizing traditional wear equations from Archard, the other is a micro-scale, material property method that using a failure criteria to determine the amount of wear. These methods are implemented at low velocities to allow for comparison to results from the literature. While the incremental method was found to provide accurate results, there are many numerical problems associated with it. The material property method was found to be feasible, but more research is needed to validate and calibrate the process.

Acknowledgements

First and foremost, I owe thanks to God for all my abilities and talents. It is only because of his grace and blessings that I have gotten this far in life.

I would also like to thank my parents for their constant encouragement and support; not only during my time at AFIT, but throughout my life. I am grateful for the prayers of all my family and friends during the writing of this thesis.

I am greatly indebted to Dr. Palazotto for his ideas and unending belief it me. I also owe thanks to my committee members: Dr. Baker and Dr. Brockman; without their advice and intervention I would still be working on this project.

Finally, I would like to thank Dr. John Schmisseur at the Air Force Office of Scientific Research for his sponsorship that has made this research possible.

Aaron Chmiel

Table of Contents

Page

List of Figures

List of Tables

List of Symbols

Symbol Page

List of Abbreviations

FINITE ELEMENT SIMULATION METHODS FOR DRY SLIDING WEAR

I. Introduction

1.1 Holloman High Speed Test Track

The Holloman High Speed Test Track (HHSTT) at Holloman Air Force Base, New Mexico, is a premier facility for high speed testing. Testing at the HHSTT has several advantages; for example, it provides cost and safety benefits over flight testing, and a more realistic environment than laboratory testing (26). Currently, the record speed achieved at the HHSTT is 2,945 m/s (6587 mph). However, there is a desire from their customers for testing at even higher speeds. In the quest for higher velocities, there are several phenomena that must be understood. One such phenomenon is wear.

Testing at the HHSTT consists of sending the test specimen down heavy duty crane rails on a rocket powered sled. Figure 1.1 shows a rocket sled train on the track. There are four stages: three pusher stages and final stage. The payload is on the top of the final stage. The sleds are held onto the track with a steel slipper that wraps around the upper portion of the rails, as shown in Figure 1.2. The rails are made of 1080 steel, and are welded together to be in constant tension when the temperature

Figure 1.1: A test sled train on the HHSTT.

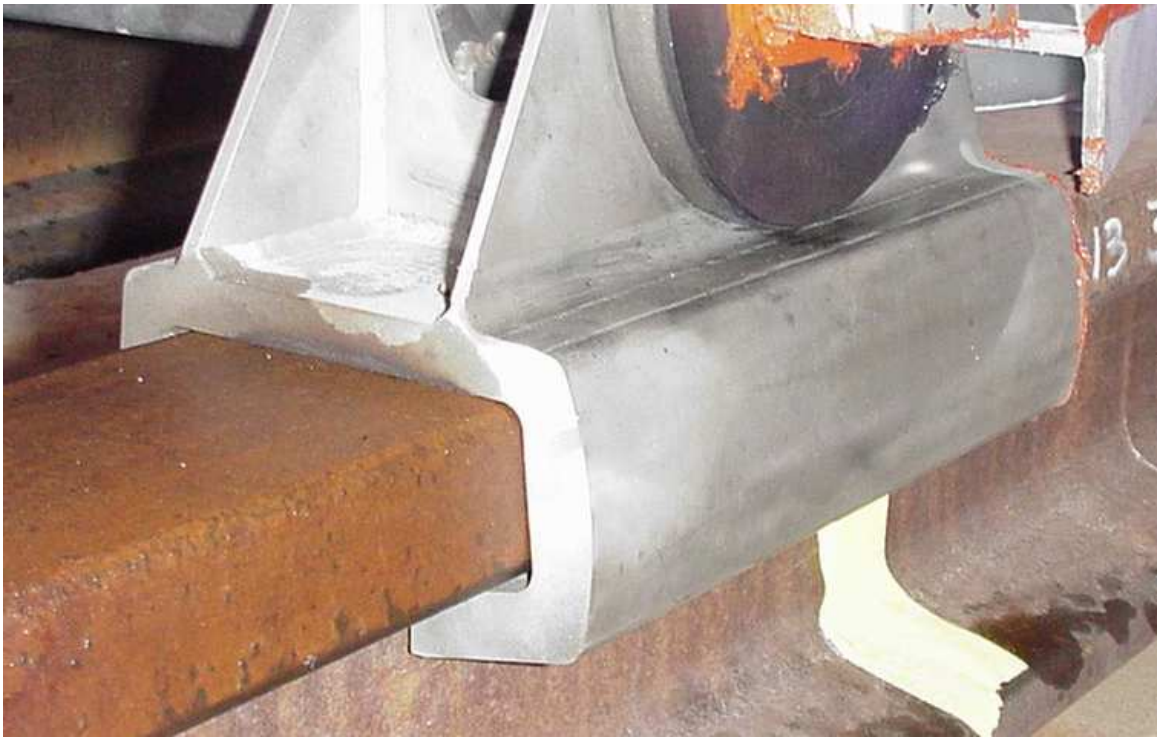


Figure 1.2: A slipper on the rail at the HHSTT

is below 120 degrees Fahrenheit. The slippers are machined out of either 4140 steel or Vascomax 300.

As the sled travels down the track, the slipper will wear. As a result of this wear, the slipper will be weakened, and the sled roll may increase (1). If there is too much wear, the slipper may not be able to hold the sled on the track, or the sled may roll excessively, resulting in a failed test. These effects of slipper wear must be taken into account when designing the slippers for a test. Currently, the methods used by the track to predict wear are simple rough estimates (1). A better wear prediction method would allow the designers to design better slippers.

1.2 Wear

Wear is a complex phenomenon that is not well understood. Humans have been aware of wear for many millennia. It has been avoided by choosing harder materials for things that are more likely to wear (23:1). However, wear is not always something to be avoided, many times it is desired; for example a pencil writes because the lead wears, and metal is polished by wearing the surface smooth. Despite the fact that the occurrence of wear has been well documented throughout history, the scientific study of wear is relatively new (14:1). Because wear is such a new field of study, there is not always agreement on theories or meanings of terms, the classifications and definitions presented in this paper are by no means the only ones that have been proposed.

The lack of understanding of wear is not merely due to it not being studied; it is an inherently difficult phenomenon to study. Because wear occurs when objects

are in contact, it is difficult to directly observe the process of wear as it happens. Therefore, experimenters generally rely on observations made after the test, and then infer what caused the wear. This is not the only difficulty; many of the causes and effects of wear only occur on a microscopic level (3; 14; 17). A third reason that wear is not well understood is that it involves many different variables that make it difficult to generalize results (14; 22).

Wear, according to Bayer, is defined as "damage to a surface as a result of relative motion with respect to another substance (3:1)." For the purposes of this work, wear shall be defined as the loss of material from a surface due to sliding along another surface. This definition is still quite broad, and encompasses many different classifications of wear. Two methods of classifying wear are: (1) the conditions in which the wear occurs; and (2) the mechanism by which the wear occurs (3:3).

Conditions used to classify wear include whether or not there is a lubricant present, and whether or not there are hard, abrasive particles present. If there is a lubricant present, it is referred to as lubricated wear, otherwise it is dry wear (14:77). If there are abrasive particles causing wear, then it is referred to as abrasive wear, otherwise it is called sliding wear (14:77). The current study focuses on dry sliding wear.

Lim and Ashby classified wear according to four mechanisms: seizure, melt, oxidation, and plasticity (17:3-4). Seizure occurs at high pressures when local asperity contacts deform until large areas of the surfaces are in contact and seize (17:11).

Melt occurs when the local temperature at the surface exceeds the melt temperature of the material and forms a thin layer of liquid (17:1). Oxidational wear occurs when a thin layer of material on the surface oxidizes and then wears away (17:13). Lim and Ashby's plastic wear encompasses several mechanisms, including adhesion of asperities, delamination and fatigue crack growth (17:17). While plasticity is an important factor in each of these, it is not the direct mechanism causing wear. Therefore, this category will be referred to as mechanical wear, following the example of Cameron (5:22). The current research focuses on mechanical wear, though the methods developed will hopefully be extensible to other wear mechanisms.

Besides classifying wear, Lim and Ashby also sorted much of the available wear data and developed equations for each mechanism. This data was mostly from pin-on-disk type wear experiments. For mechanical wear, they used Archard's wear law (17:17):

$$\tilde{w} = K\tilde{p} \tag{1.1}$$

Where

$\tilde{w} =$ non-dimensional wear rate

$\tilde{p} =$ non-dimensional pressure

$K =$ Archard's wear coefficient

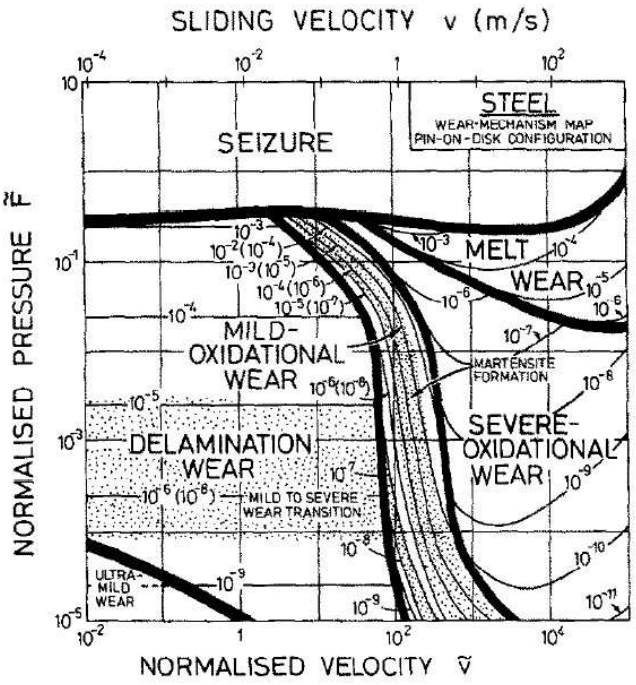


Figure 1.3: Lim and Ashby's wear map for steels

The parts of this equation will be discussed in more detail in Chapter 2. Archard's wear law has also been used by many other researchers to model mechanical wear (11; 18; 25).

Lim and Ashby created contour plots of wear rate as a function of normalized pressure and normalized velocity using the equations they developed. They then combined the plots for each mechanism onto a single graph showing where each mechanism dominates. This type of graph they called a wear map. A wear map for steel is shown in Figure 1.3. Oxidational wear was divided into two parts, mild and severe. Mechanical wear is in the region labeled "delamination wear".

As finite element analysis (FEA) has become more accessible, many researchers have looked for ways to use it to calculate wear. Each has their own unique details,

but the general formula is to alternate between a finite element analysis to determine pressures and a calculation of wear which adjusts the model (4; 11; 20). Podra and Andersson say that this method is best suited to the comparison of different design options due to the modeling simplifications and the uncertainty in the input data (20: 81). Molinari et al. show that a finite element based wear prediction method can be calibrated to provide accurate results (18:606). Benabdallah and Olender use a similar method to determine the profile generated on a pin in a pin-on-disk wear experiment, and produced good agreement with experiment (4:1223). They also found that the wear profile eventually reaches a steady state condition, where any subsequent wear is distributed evenly across the surface (4:1223).

The interaction between the rail and the slipper at the HHSTT was studied extensively in the context of gouging. The hydrocode CTH was used to model the material interaction between the slipper and rail that causes gouging (6; 16; 24). Greg Cameron modified the model and methodology used to study gouging and applied it to the study of wear. He used the hydrocode to determine stress and temperature of a slipper in situations that may result in wear and applied a criterion to each property to calculate a wear depth (5:63). He found that it was necessary for the slipper to run into an asperity for wear to be predicted (5:84). Additionally, Cameron used Lim and Ashby's equations for melt wear and mechanical wear to predict wear at the HHSTT. Using these equations he got reasonable answers, but did not compare them with actual experimental results.

1.3 Problem Statement and Chapter Overview

The goal of this research is to find a method to predict the wear that a slipper on the HHSTT will experience during a test run. The particular difficulty of this problem comes from the extremely high speeds attained at the track. Velocities in previous laboratory wear experiments have rarely exceeded more than a few hundred meters per second, but the sleds at the track greatly exceed this speed (4;17;19;20). These low speed equations cannot be used at higher velocities without verification that they are applicable. If the equations do apply to high velocities, they still require experimental data to calibrate each test. Therefore, a goal of this research is to develop a method that can be used to predict wear through a finite element analysis at any velocity without relying as heavily on equations developed by laboratory experiments.

Two methods are investigated. The first is an incremental, macro-scale approach that follows previous work to predict wear with finite elements, especially that done by Benabdallah and Olender. This is done by means of a script that calculates wear according to Archard's wear law and finite element analysis outputs. This method is set aside, because it has significant numerical problems and is dependent on equations for wear developed at low velocities. The second, a micro-scale approach, is then taken which is similar to the material property method used to study gouging with CTH, but using finite element analysis. This method is developed to the point that a proof of concept can be carried out. The calibration and validation of the method require further study, and are beyond the scope of this work. It is possible that the results

from the micro-scale approach could be incorporated in a macro-scale approach similar to the first.

Some background information and theory is presented in Chapter 2. The development of the incremental wear method is in Chapter 3, and some results are discussed in Chapter 4. The development and methodology of the material property wear method are described in Chapter 5. Results from a proof of concept for this method are presented and discussed in Chapter 6. Finally, Chapter 7 concludes the document and contains some points for further study. Appendices contain the source code for the wear script described in Chapter 3 and step by step results for the incremental method.

II. Theory and Tools Used

2.1 Abaqus

For this research, the commercial FEA suite Abaqus was used. Abaqus was chosen for its excellent non-linear capabilities, and because it had been used by the researchers before and so did not require further investment in procurement or training. The Abaqus suite has several parts, but only three were used in the current study: Abaqus/CAE, Abaqus/Standard, and Abaqus/Explicit. Abaqus/CAE is a graphical user interface (gui) for pre-processing and post-processing a finite element analysis, Abaqus/Standard and Abaqus/Explicit are two different analysis solvers. These three parts are briefly described below.

Abaqus/CAE is a graphical tool which allows an analyst to create and prepare a model for analysis and then view the analysis results. Abaqus/CAE provides an analyst with tools to create a geometric model of the structure to be analyzed, give the model material properties and a mesh, and to setup an analysis in a way which allows the analyst to make small changes quickly without much hassle. Once an analysis has been setup, a file is written which is input to either Abaqus/Standard or Abaqus/Explicit. The solvers can write results to either a binary output database, or a plain text file. The output database can be viewed in Abaqus/CAE, where pictures and graphs can be created.

Abaqus/Standard and Abaqus/Explicit are the two solvers in the Abaqus suite. The actual algorithms used are proprietary, so they cannot be detailed here, but the main features and intended uses of each will be discussed. Both solvers can handle nonlinearities in the materials, loads, and geometry. This is one of the primary strengths of Abaqus. However, each has its own uses, and particular features, so it is up to the analyst to decide which is best suited to his particular problem. This dichotomy of solvers can however, be a source of frustration when the analyst desires to use two features that are not in the same solver.

Abaqus/Standard is the general purpose solver for Abaqus. It is used for many different types of analyses, including static, dynamic and eigenvalue problems. The solution methods for dynamic problems are unconditionally stable, that is, the time step (Δt) is dictated by the accuracy of the results not mathematics. Abaqus/Standard also has many user subroutines available, which allow the user to customize the functionality of the solver to better meet his needs. Abaqus/Explicit is a more specialized solver that focuses on time-dependent wave propagation problems. It uses a conditionally stable, explicit, direct time integration scheme to solve the time-dependent equation of motion. The maximum time step for stability is determined by the Courant-Freidricks-Lewy condition, which is

$$\Delta t \leq \frac{\mathrm{L}}{\mathrm{c}}$$

where L is the smallest element dimension and c is the speed of sound in the material (7). Abaqus/Explicit has more material property models, but fewer user subroutines than Abaqus/Standard. For a more detailed discussion of the workings of Abaqus, and the differences between the two solvers, the reader is referred to the Abaqus users manual (9).

2.2 DADS

The test designers at the Holloman High Speed Test Track use a commercial, off-the-shelf program called the Dynamic Analysis and Design System, or DADS, to predict the loading of a sled during a test. Among the data reported by DADS are the sled velocity, and forces on the slippers as functions of time. The forces on the slippers are recorded at several locations around the cross-section, these locations are shown in Figure 2.1. For this study DADS predictions for test mission 80X-1* were used for various values. The vertical force at one location is shown in Figure 2.2 as a sample of the what the output from DADS looks like. This test reached a maximum velocity of 2,885 m/s. For this sled, the slippers were 20.32 cm long, 2.54 cm thick and 10.16 cm wide with a 0.3175 cm gap (nominal) between the shoe and the rail (13). The slippers were made of Vascomax 300.

2.3 Pin-On-Disk

There are many different ways to set up a wear experiment depending of the purpose for the experiment. A popular set up is the pin-on-disk. In pin on disk

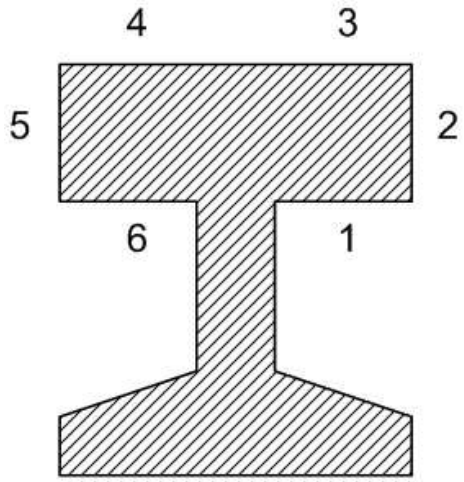


Figure 2.1: Cross-section of the rail showing the locations where DADS reports forces on the slippers (2)

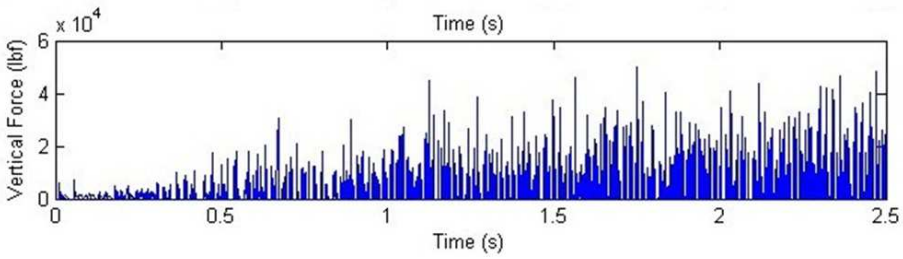


Figure 2.2: A plot of the vertical force data predicted by DADS for one slipper location through the entire run

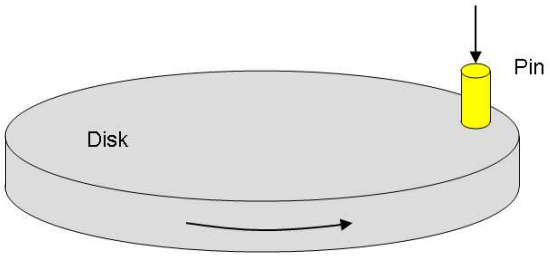


Figure 2.3: A typical pin-on-disk experimental set up

experiments, a small pin is pressed against a flat disk. While there are standards for pin-on-disk experiments, there is also a large amount of variety in many of the parameters from one experiment to the next (3). The velocity, loading and geometry can all be varied in order to more closely simulate the problem of interest. Figure 2.3 shows a cartoon of a typical pin-on-disk experimental set up.

2.4 Wear Parameters

There are several parameters that are used to quantify wear. The most important are: wear volume (V), wear height (h), wear rate (w) and wear coefficient (K). Wear volume is simply the volume of material that is lost due to wear. It is often measured by comparing before and after volumes of wear specimens. Wear height can be either the maximum, average, or minimum depth of the material removed; depending on what is most important to the experimenter. These two measurements are often what is of most interest to engineers and designers, who are concerned with the life and durability of a part. They are not, however, very useful for characterizing the process of wear in general terms, because they are heavily dependent on variables that may change significantly from problem to problem. Some of these variables are: the

distance traveled (s), the time over which the wear occurred (t), the normal force at the interface (F_N), and the hardness material that is wearing (H). Oftentimes, the distance and time are combined as velocity (v), and force is combined with area as pressure (P).

Wear rate is a more general parameter for specifying wear. It is not a time rate, but a distance rate. Wear rate is defined as the volume removed per sliding distance, that is:

$$w = \frac{\mathrm{V}}{\mathrm{s}} \tag{2.1}$$

The distance traveled can be determined by multiplying the time that the object was in motion by the velocity at which it was traveling. The wear rate can be non-dimensionalized for comparison with different size specimens. The non-dimensional wear rate ($\tilde{w}$) is obtained by dividing by the apparent contact area (A)

$$\tilde{w} = \frac{\mathrm{V}}{A\mathrm{s}} \tag{2.2}$$

Wear rate, either with or without dimensions, is generally considered to be constant for a material pair. This parameter is a function of only pressure and velocity, which makes it good for calculation with a single set of materials.

The problem can be further non-dimensionalized by using the wear coefficient and normalized force ($\tilde{P}$) and velocity ($\tilde{v}$). The force is normalized by:

$$\tilde{P} = \frac{F_n}{AH} = \frac{P}{H} \tag{2.3}$$

Because the force is normalized by the area, it can also be referred to as normalized pressure. Velocity is normalized by the hardness and apparent contact area:

$$\tilde{v} = \frac{vr_o}{a} \tag{2.4}$$

where r_o is the radius of the pin, and a is the thermal diffusivity of the material in m^2/s (5:20). The wear coefficient comes from Archard's wear law (20:72):

$$\tilde{w} = \mathrm{K}\tilde{P} \tag{2.5}$$

where k_A is Archard's wear coefficient:

$$\mathrm{K}_A = \frac{\mathrm{V}H}{\mathrm{s}F_N} \tag{2.6}$$

The wear coefficient is intended to allow for the results from known experiments to be applied to predict wear for new geometries or materials. However, in practice it needs to be determined experimentally for each contact configuration (20). This is because wear is not simply a material property, but a response of a system (3:v) The unreliability of the wear coefficient is currently one of the major problems with wear predictions based on Archard's wear law. Because it may change with each configuration, the accuracy of any equations that use it is reduced. This research intends to address this shortcoming.

2.5 Damage Initiation Criteria (9; 12)

Abaqus/Explicit includes several models to predict the initiation of damage, or failure, in a material. The model chosen for this study was developed by Hooputra, Gese, Dell, and Werner for crashworthiness simulation with finite elements. Their model contains failure criteria for both ductile and shear fracture. Ductile fracture is caused by the nucleation and coalescence of of microstructural voids. Shear fracture is caused by the localization of shear bands. Both criteria define the equivalent plastic strain at fracture (ε_D^{**},ε_S^{**}) as a function of stress triaxiality (η) which is:

$$\eta = \frac{3\sigma_m}{\sigma_{eq}} = \frac{\sigma_1 + \sigma_2 + \sigma_3}{\sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2 - \sigma_1\sigma_2 - \sigma_2\sigma_3 - \sigma_3\sigma_1}} \tag{2.7}$$

where

$$\sigma_m = \frac{\sigma_1 + \sigma_2 + \sigma_3}{3}$$
$$\sigma_{eq} = \sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2 - \sigma_1\sigma_2 - \sigma_2\sigma_3 - \sigma_3\sigma_1}$$

and σ_1, σ_2, σ_3, are the principal components of the stress tensor.

The equivalent plastic strain at ductile failure is:

$$\varepsilon_D^{**} = \frac{\varepsilon_T^+ \sinh[c(\eta^- - \eta)] + \varepsilon_T^- \sinh[c(\eta - \eta^+)]}{\sinh[c(\eta^- - \eta^+)]} \tag{2.8}$$

where η^+ and η^- are the stress triaxiality for equibiaxial tension and compression, ε_T^+ and ε_T^+ are the equivalent fracture strains in equibiaxial tension and compression,

respectively, at ductile failure, and c is an orientation dependent parameter. To use this criterion in Abaqus/Explicit requires curves of fracture strain versus stress triaxiality for relevant strain rates in tabular form (8). The ductile failure risk parameter (ω_D) is used to determine when damage is initiated. Ductile damage is initiated when:

$$\omega_D = \int \frac{\mathrm{d}\varepsilon^{**}}{\varepsilon_D^{**}} = 1$$

For the shear criterion, the equivalent plastic strain at failure is:

$$\varepsilon_S^{**} = \frac{\varepsilon_S^{+}\sinh[f(\theta - \theta^{-})] + \varepsilon_S^{-}\sinh[f(\theta^{+} - \theta)]}{\sinh[f(\theta^{+} - \theta^{-})]} \tag{2.9}$$

where

$$\theta = \frac{1 - k_s\eta}{\phi}$$

$$\phi = \frac{\tau_{max}}{\sigma_{eq}}$$

θ^{+} and θ^{-} are the values of θ for equibiaxial tension and compression, ε_S^{+} and ε_S^{+} are the equivalent fracture strains in equibiaxial tension and compression, respectively, at shear failure, k_s is a material parameter and f is an orientation dependent parameter. θ is called the shear stress ratio. To use this criterion in Abaqus/Explicit requires curves of fracture strain versus shear stress ratio for relevant strain rates in tabular form and a value for k_s (8). The shear failure risk parameter (ω_S) is used to determine

when damage is initiated. Shear damage is initiated when:

$$\omega_S = \int \frac{\mathrm{d}\varepsilon^{**}}{\varepsilon_S^{**}} = 1$$

Hooputra et al. describe a process for finding the required parameters. They also validate their model for three-point bending and axial compression of a double chamber aluminum extrusion. For more information on the failure model, the reader is referred to (12) and (9).

III. Incremental Method Development

The incremental method of predicting wear is a macro-scale approach based on experimentally determined wear equations, and has been investigated by several other researchers (4;20;25). This method has been studied for wear at low speeds for pin-on-disk type experiments, but it has not been applied to high speed situations. Briefly, it consists of iterating between using a finite element analysis to find pressures and using Archard's wear equation to modify the model to reflect the amount it has worn. This chapter outlines the development of a script that applies this method in Abaqus and a model to test it on.

3.1 Model

The model used during development of the incremental method is a plane strain model. It was based on the pin-on-disk experiments done by Benabdallah and Olender; there are some differences, but they will not be detailed. The script is intended to be independent of the model used so long as certain restrictions on the mesh arrangement are met. These restrictions will be discussed in a later section.

The pin is 3.0 mm across at the bottom for 2.54 mm, and then widens to 4.76 mm for another 5.52 mm at the top part of the pin. Figure 3.1 shows the dimensions of the pin. Along the bottom of the pin the elements are 6×10^{-3} mm square. This is section called the wear region because it is the region of the mesh that will be adjusted when the pin wears. Above the wear region is a transition region consisting of two

Table 3.1: Material properties of Vascomax 300 used in the pin model (6:5-11)

Young's Modulus	Poisson's Ratio	Density
180.7 GPa	0.283	8000 kg/m^3

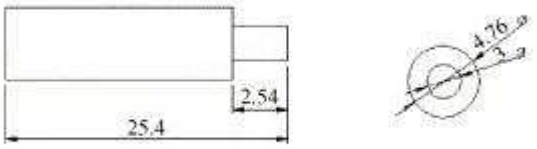
rows of triangular elements that transition the mesh from the wear region to the rest of the pin where the elements are four times larger (nominally). Figure 3.1 shows the mesh used on the pin. The pin has 1236 elements, with a total of 7526 degrees of freedom. The pin has the material properties of Vascomax 300 as determined by Cinnamon, which are shown in Table 3.1 (6:5-11).

The disk is modeled with an analytical rigid surface. This reduces the degrees of freedom in the model. Benabdallah and Olender also modeled the disk with a rigid surface.

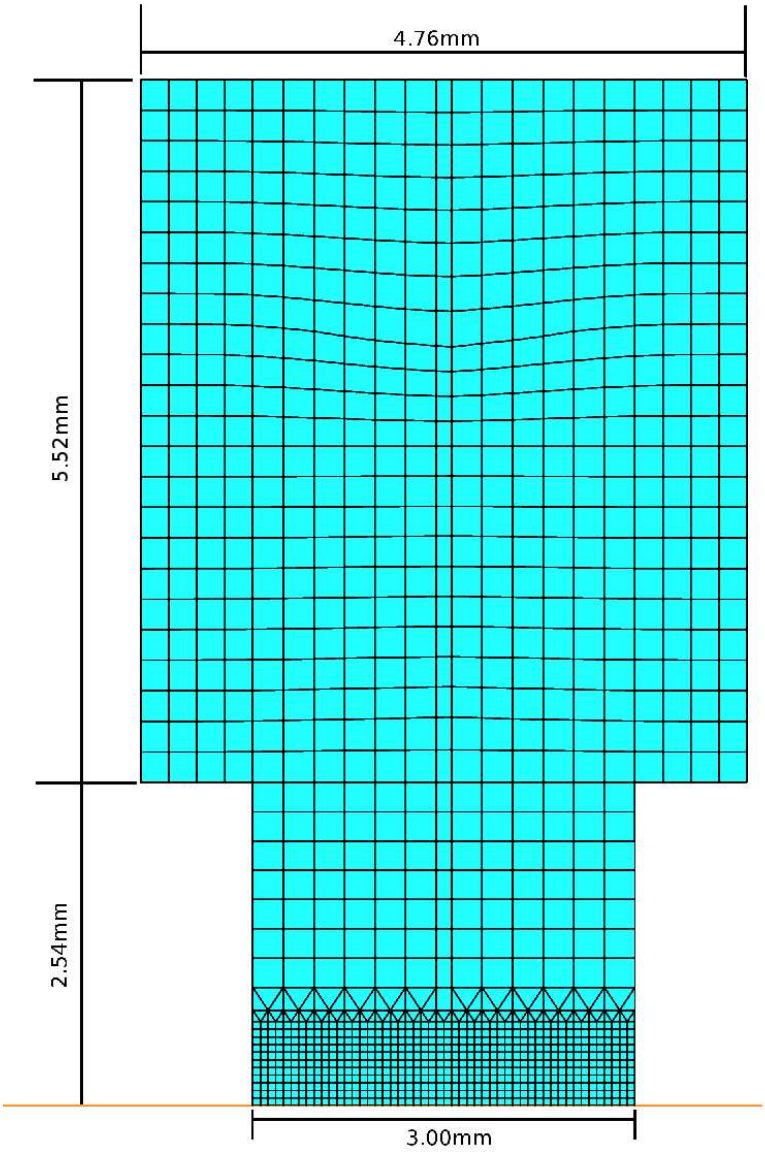
Figure 3.2 shows the loads and boundary conditions for the model. The pin is pressed against the rail by a uniform pressure load on the top surface. The top half of the sides of the pin are constrained in the x direction so that the pin can only move vertically. The disk is restrained in the y direction and is moved to the right with a constant velocity boundary condition.

3.2 Mesh Validation

The validity of the mesh was checked by comparing results from analyses using different element types or mesh refinements to known analytical solutions. The bottom portion of the pin was modeled as a 2.54 cm tall by 3.0 cm wide rectangle. The same mesh as on the pin was used, but by eliminating the top portion, the number of



(a) Pin used by Benabdallah and Olender



(b) Abaqus model of the pin

Figure 3.1: The pin and associated model, showing the dimensions and the mesh

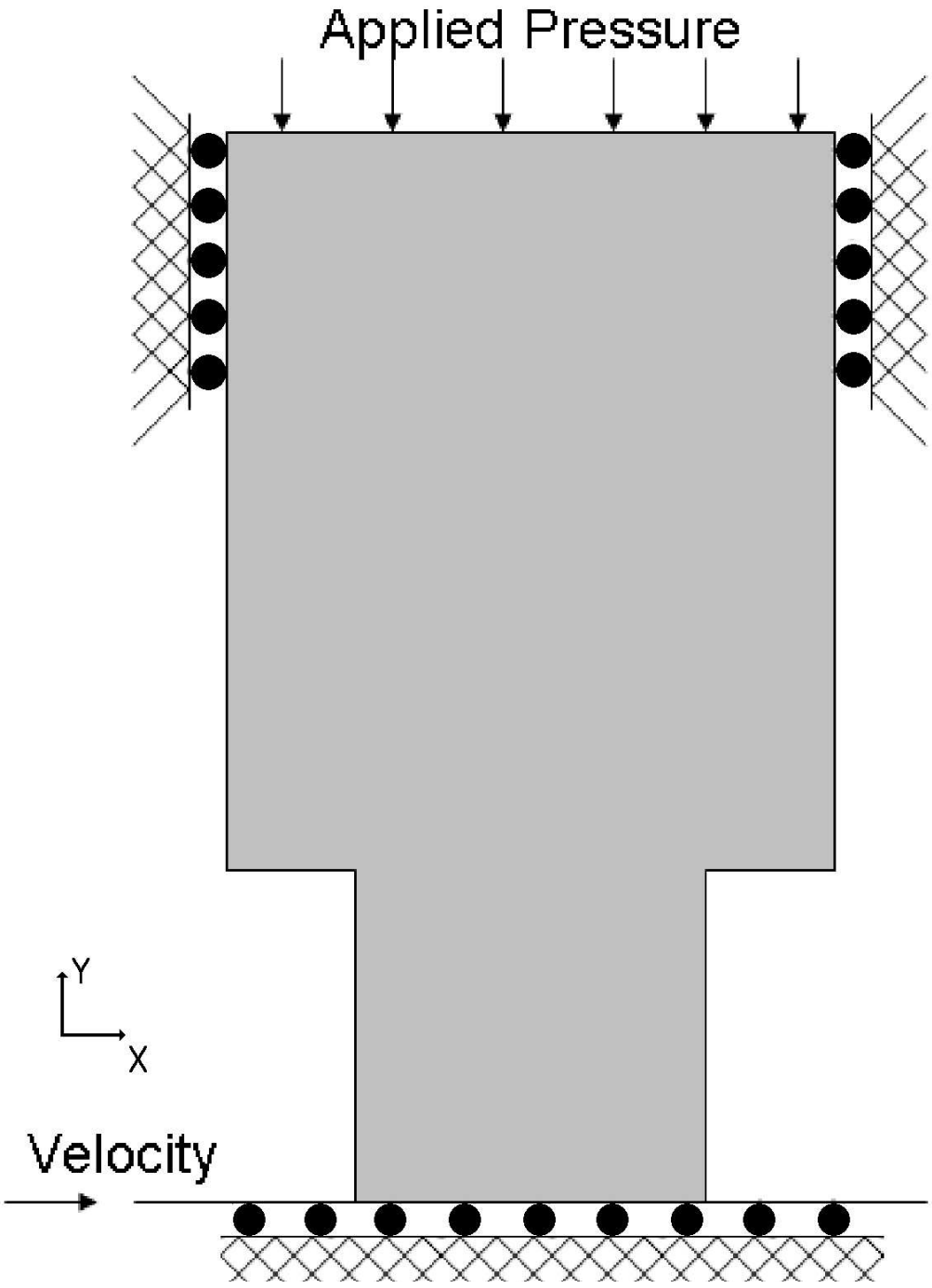


Figure 3.2: The boundary conditions applied to the pin used for the incremental method

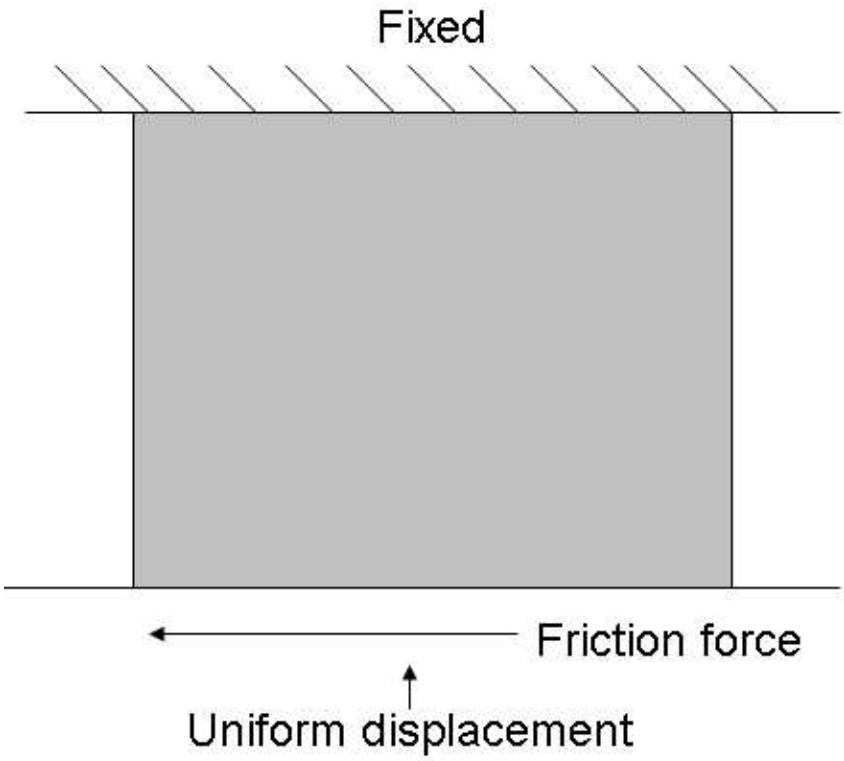


Figure 3.3: Boundary conditions for the mesh validation model

degrees of freedom was reduced to 4200. The rectangle was fixed at the top and the sides were free boundaries. The analytical rigid surface was slid against the bottom of the rectangle. The rectangle was compressed by displacing the rigid surface into the block. This simplified geometry was used so that the results could be compared to the analytical solution developed by Prasad et al. The boundary conditions for the mesh validation model are shown in Figure 3.3.

Curves of normal stress along the bottom were compared to each other and to the shape predicted by Prasad, Chiu, and Dasgupta, which is shown in Figure 3.4. This figure depicts the normal stress at the leading edge having an extremely high magnitude, which drops quickly and levels out in the middle before dropping again

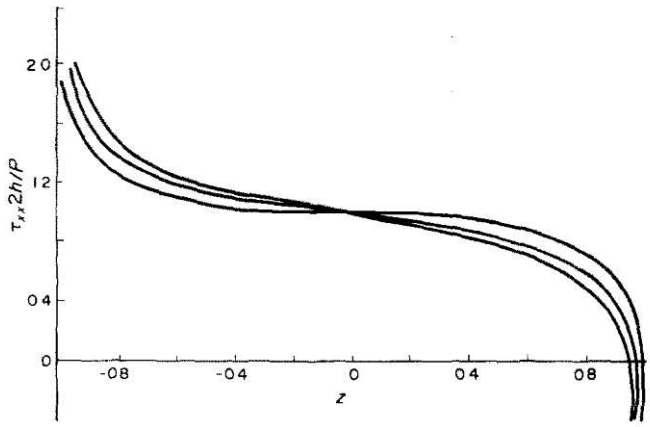


Figure 3.4: Variation of contact pressure predicted by Prasad et al. (21:629). The curves are for 3 different aspect ratio rectangles

to zero near the trailing edge. The shear stress along the free edge ought to be zero. Curves of shear stress along the free edge were plotted to see how close the shear stress was to zero.

Elements with either linear or quadratic shape functions and either full or reduced integration schemes were considered. This resulted in four element types to be compared:

- Linear elements with full integration (Linear-Full)
- Linear elements with reduced integration (Linear-Reduced)
- Quadratic elements with full integration (Quadratic-Full)
- Quadratic elements with reduced integration (Quadratic-Reduced)

The normal stress distribution for all element types matched the shape predicted by Prasad et al., as can be seen by comparing Figure 3.5 to Figure 3.4. Note that the values on the axes are not the same in the figures because Prasad et al. plotted

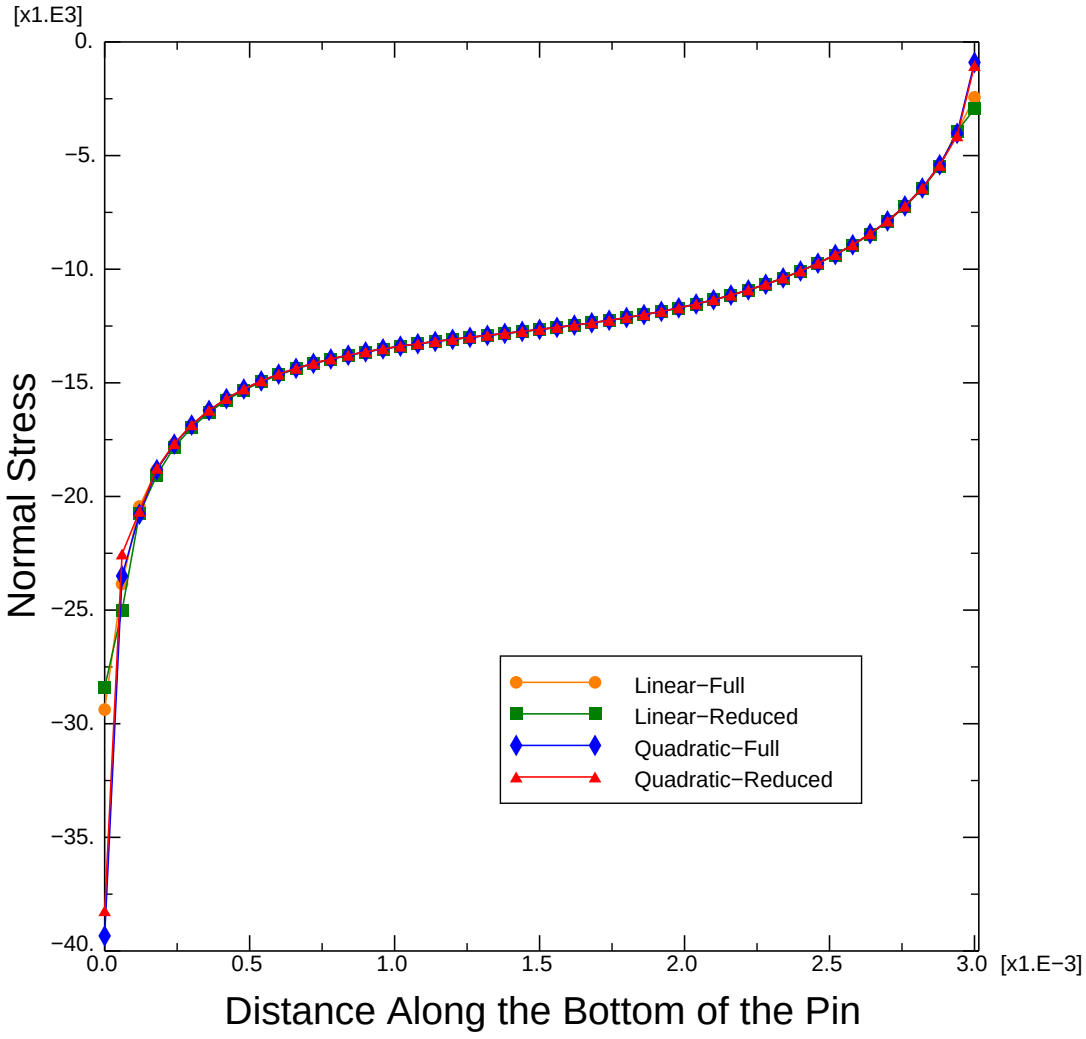


Figure 3.5: Normal stress along the bottom surface of the validation model for different element types

normalized stress and normalized distance in Figure 3.4, but the values in Figure 3.5 are actual values. The difference from one element type to another was not large, as can be seen by comparing the various curves in Figure 3.5. Figure 3.6 shows the shear stress along the trailing free edge. The quadratic elements provided shear values much closer to zero than the linear elements. The difference between full integration and reduced integration was less significant. Quadratic elements with full integration were chosen because they were the best at providing zero shear stress at the free edge. Reduced integration quadratic elements could be used to reduce computational cost if necessary.

The refinement of the wear region was also studied to see if smaller elements would produce a better prediction of the contact pressure distribution along the suface. Benabdallah and Olender used elements 6×10^{-5} m square in the wear region of their model. Elements this size were compared with elements half this size. The refined mesh is shown in Figure 3.7. Larger elements were not considered because they would reduce the resolution of the wear profile in the final test. Figure 3.8 shows that both meshes produce the same pressure distribution. Thus, halving the element size does not produce significant gains in the accuracy of the solution. Therefore, the larger, 6×10^{-5} m, square elements were used in the wear region of the final model.

3.3 Analysis Type Determination

In their study, Benebdallah and Olender used a dynamic analysis, but Podra and Andersson used a static analysis. A comparison of quasi-static analysis versus

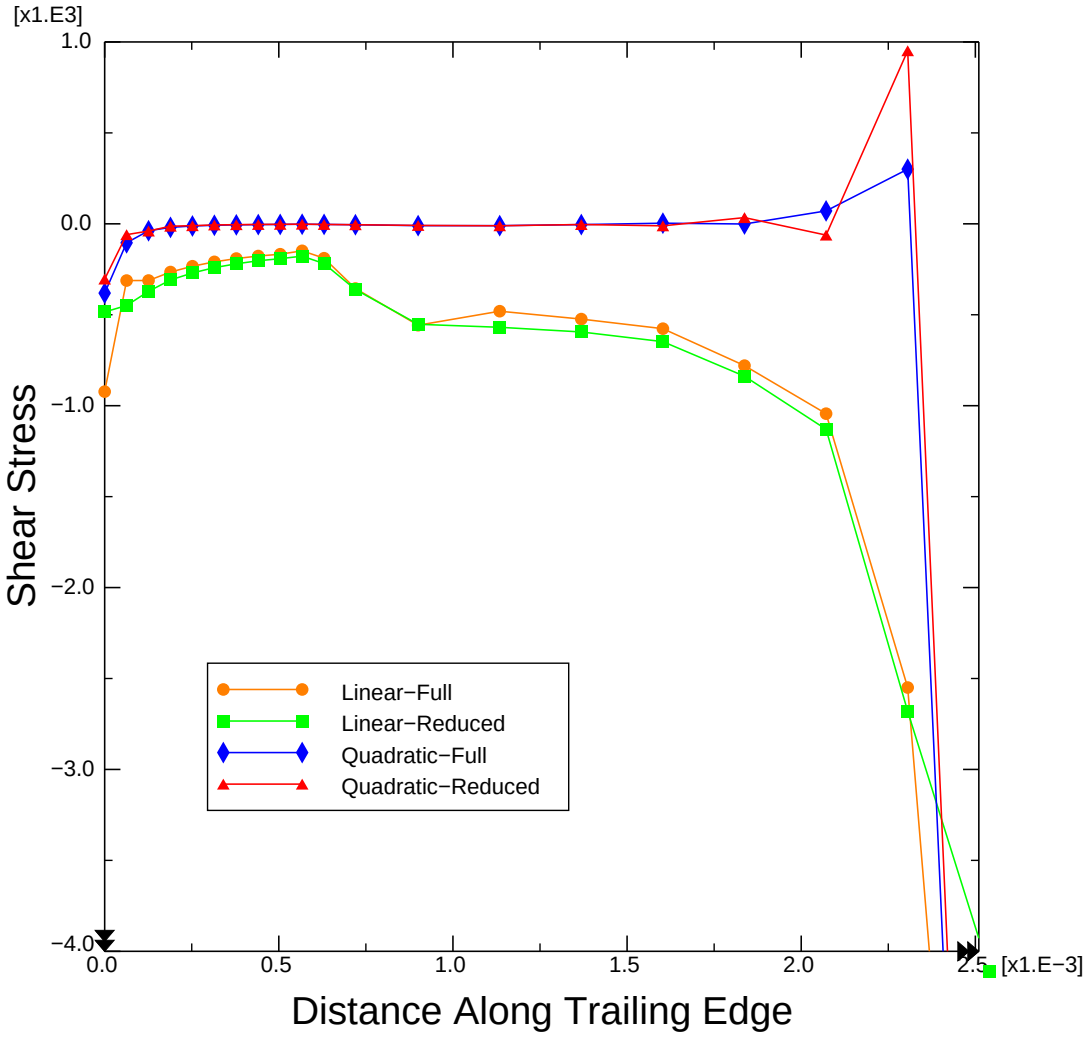


Figure 3.6: Shear stress along the trailing edge of the validation model for various element types

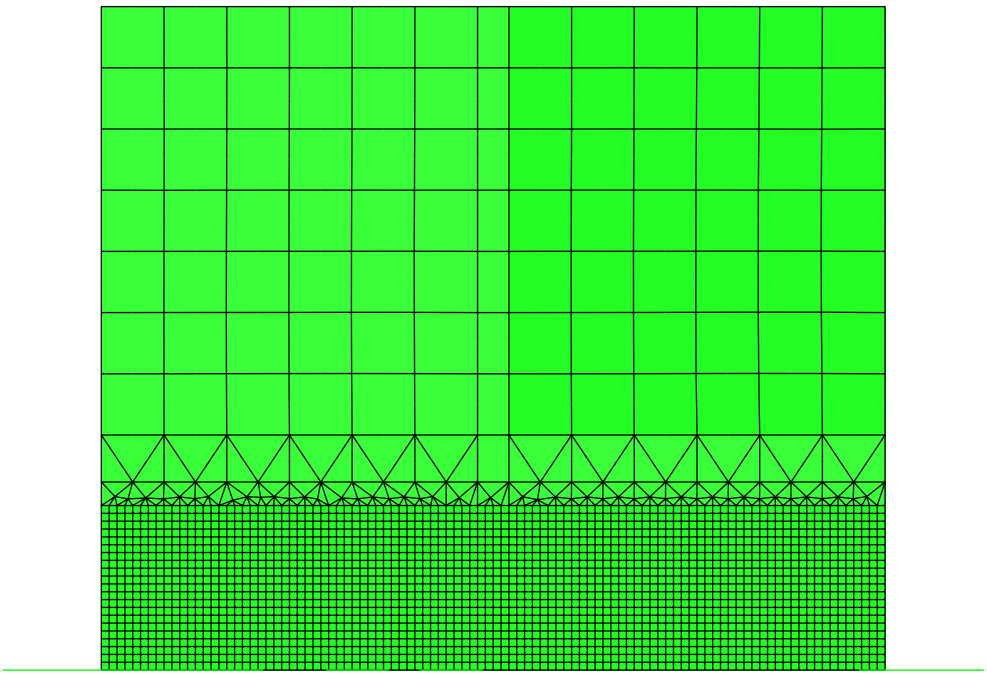


Figure 3.7: The refined mesh used for the mesh convergence study

dynamic analysis was performed on the full model to determine which was best for the current problem. Other than the analysis type, nothing in the model was changed, the same loads and boundary conditions were used for both analyses. Both of these analyses were performed with Abaqus/Standard. The comparison was based on two criteria: the time required for the analysis, and the accuracy of the pressure distribution along the bottom of the pin. Figure 3.9 shows the distribution of pressure for both analyses. The curves are nearly indistinguishable, thus the analysis type has no effect on the pressure distribution. This is partially due to the massless disk being moved while the pin is stationary. If the pin was moved and the disk stationary there would be inertial effects due to the mass of the pin. The quasi-static analysis took

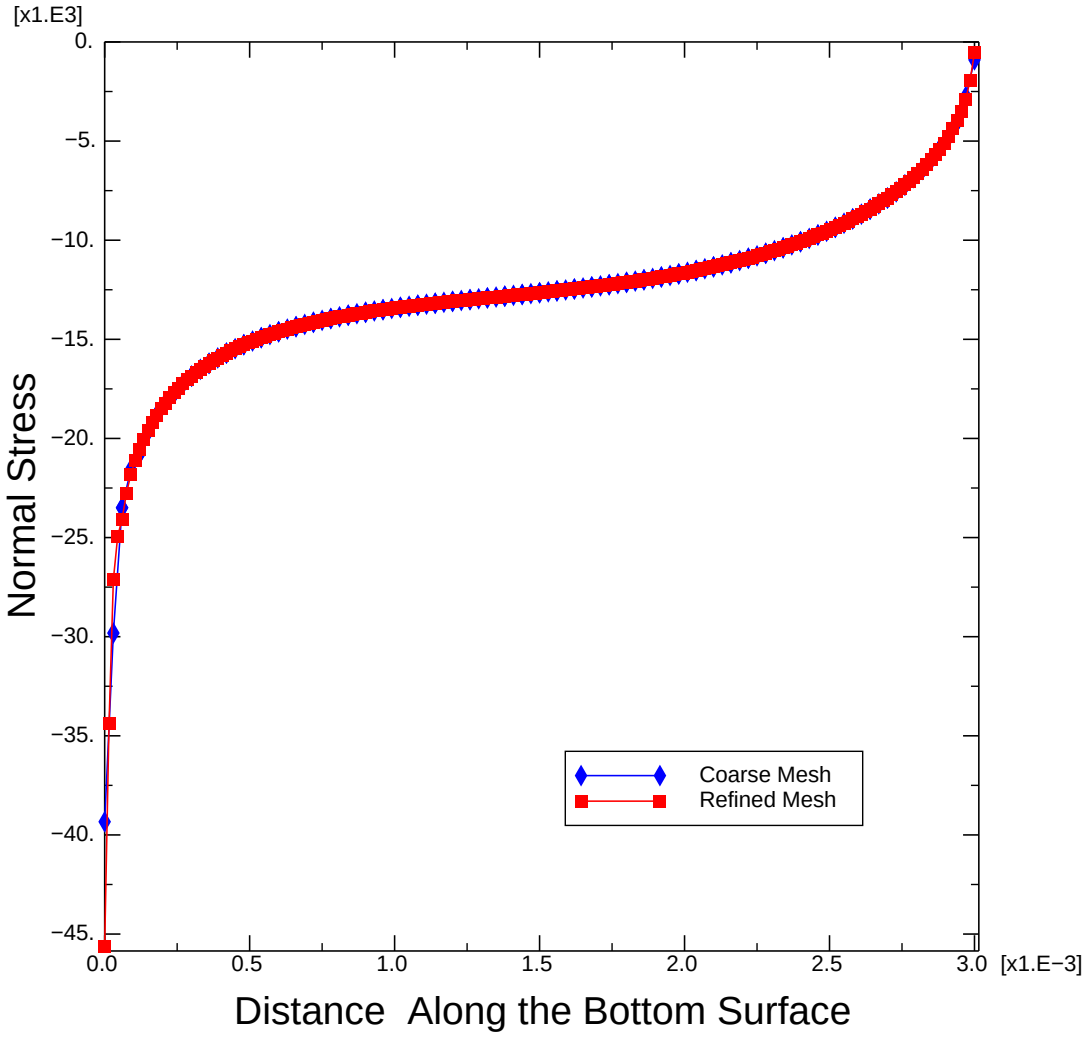


Figure 3.8: Normal stress along the bottom surface of the validation model for two different element sizes

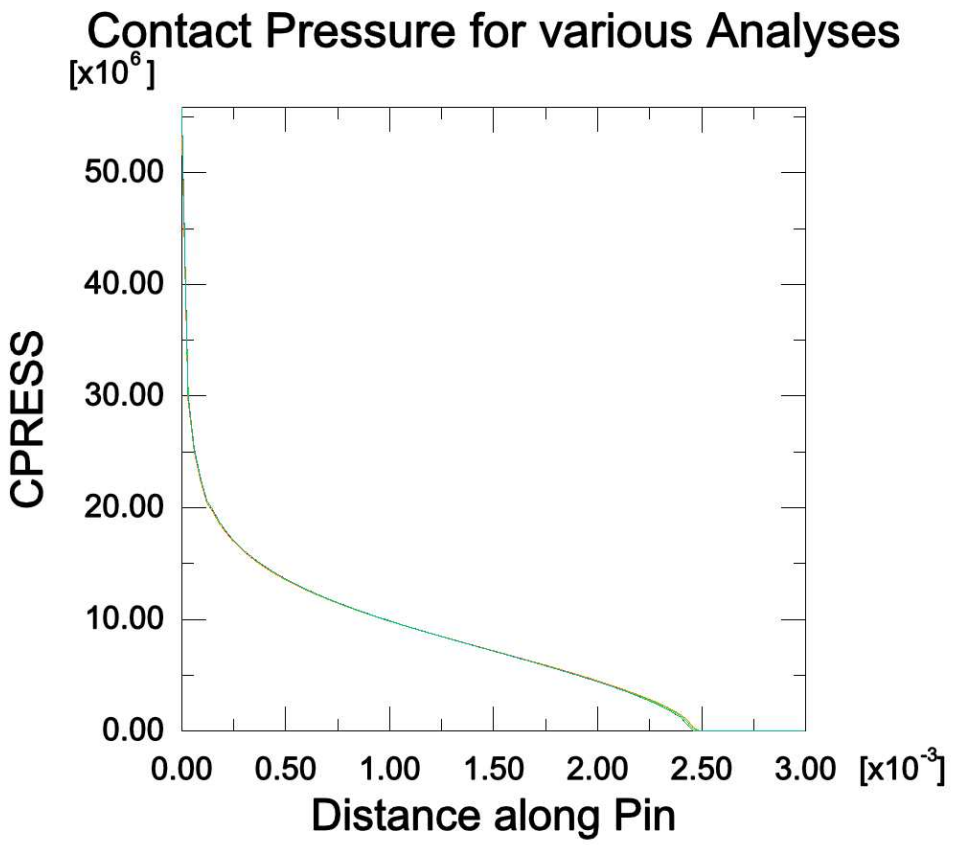


Figure 3.9: Comparison of contact pressure for dynamic and quasi-static analysis

significantly less time to complete than the dynamic analysis. Therefore, quasi-static analyses were used in the incremental method.

3.4 Friction

A test was also performed to determine the effect that a change in the velocity of the disk would have on the pressure distribution. The pressure distribution along the bottom of the pin was compared for four disk velocities: 0 m/s, 1 m/s, 10 m/s and 100 m/s. All other parameters were the same for each analysis. A plot of the pressure distribution along the bottom of the pin for all four analyses is shown in

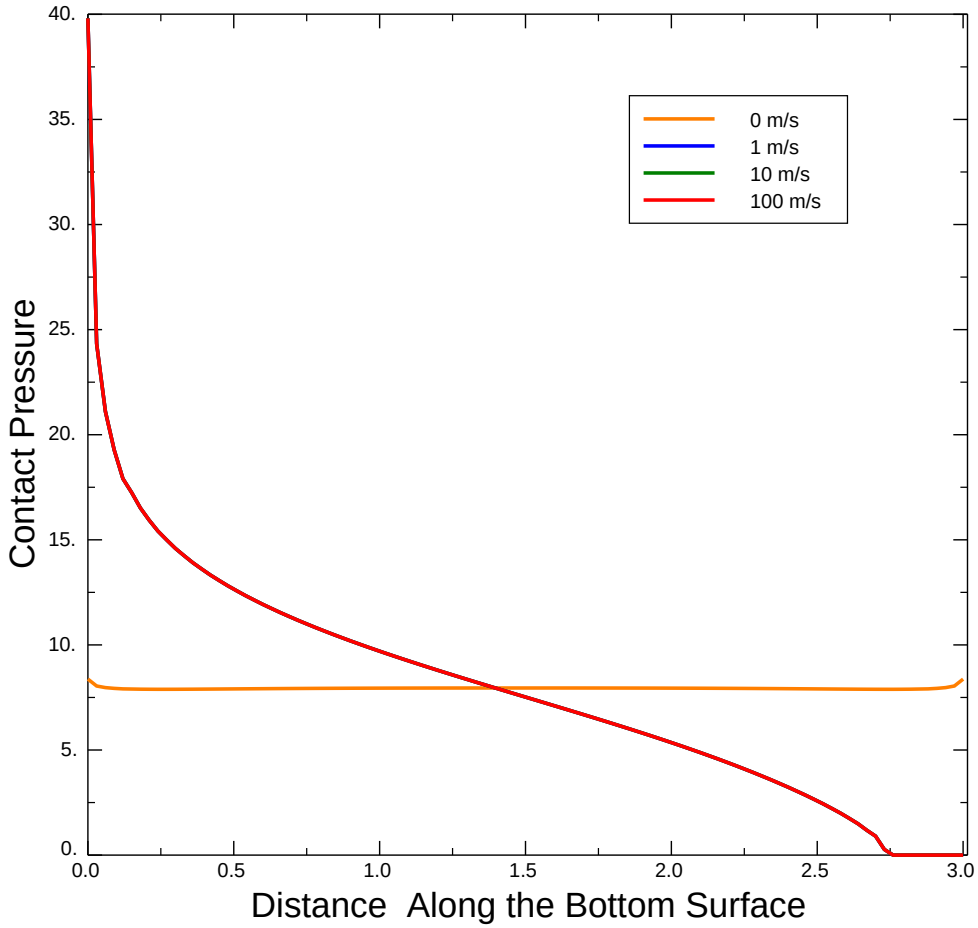


Figure 3.10: Pressure distribution along the bottom surface of the pin for different velocities

Figure 3.10. For the test with no velocity, the pressure was nearly uniform, with only a slight turn upward at the corners of the pin. The other three cases all show the same distribution as predicted by Prasad, with no deviation between the three. This indicates that a change in velocity will not change the pressure distribution. This was expected because there was no velocity dependence in the model. In reality the coefficient of friction changes with velocity. A velocity dependent coefficient of friction was added into the analysis to remedy this discrepancy.

In Abaqus, the coefficient of friction is used to introduce the shear due to friction of sliding to the problem. The shear stress is obtained by muliplying the normal stress at the sliding surface by the coefficient of friction. Cameron developed the following equation for the coefficient of friction as a function of the product of normalized pressure ($\tilde{P}$, equation 2.3) and normalized velocity ($\tilde{v}$, equation 2.4) based on data from pin-on-disk wear experiments by Montgomery (5):

$$\mu = 1.5704(\tilde{P}\tilde{v})^{-0.2299} - S \tag{3.1}$$

where

$$S = \min\{1.5704(\tilde{P}\tilde{v})^{-0.2299}\} - 0.02$$

Values for this equation were tabulated for pressures from 1 MPa to 20 MPa in 1 MPa steps, and velocities from 0.01 m/s to 10 m/s in 0.01 m/s steps. The following values were used to normalize the pressure and velocity:

$$A = 2.8273 \text{ m}^2$$

$$H = 1000 \text{ MPa}$$

$$r_o = 0.003 \text{ m}$$

$$a = 4.5 \text{ m}^2/\text{s}$$

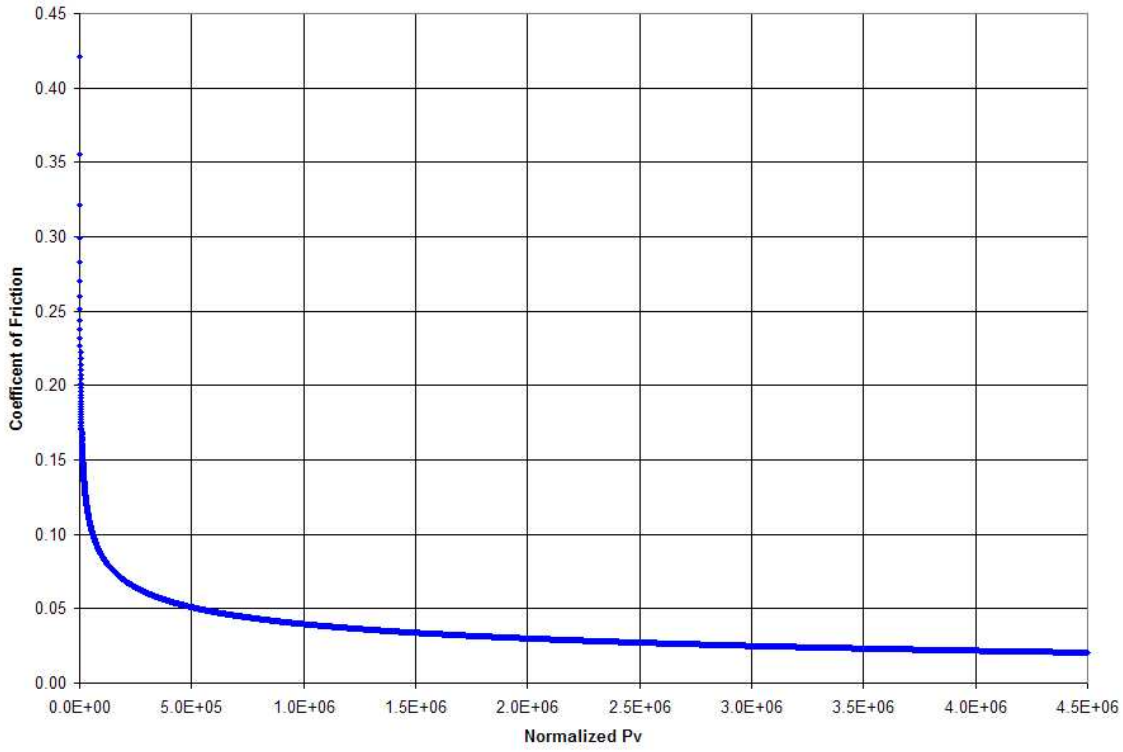


Figure 3.11: Coefficient of friction equation plotted against $\tilde{P}\tilde{v}$

The values for A and r_o are from the geometry, and the value for H and a were taken from Cameron's values for Vascomax 300 (5:37). The tabulated data was then added to the model. Figure 3.11 shows a plot of the coefficient of friction versus $\tilde{P}\tilde{v}$.

3.5 Corner Singularities

The upward turn at the ends of the contact pressure curve for zero velocity in Figure 3.10 is due to the square corners on the pin. Sharp corners, like those on the pin, tend to be concentration points for stress, and analytically, a singularity is predicted at these points. Due to the approximate nature of FEA, a high, yet finite, pressure is predicted. In reality, there are no sharp corners; they are all rounded off

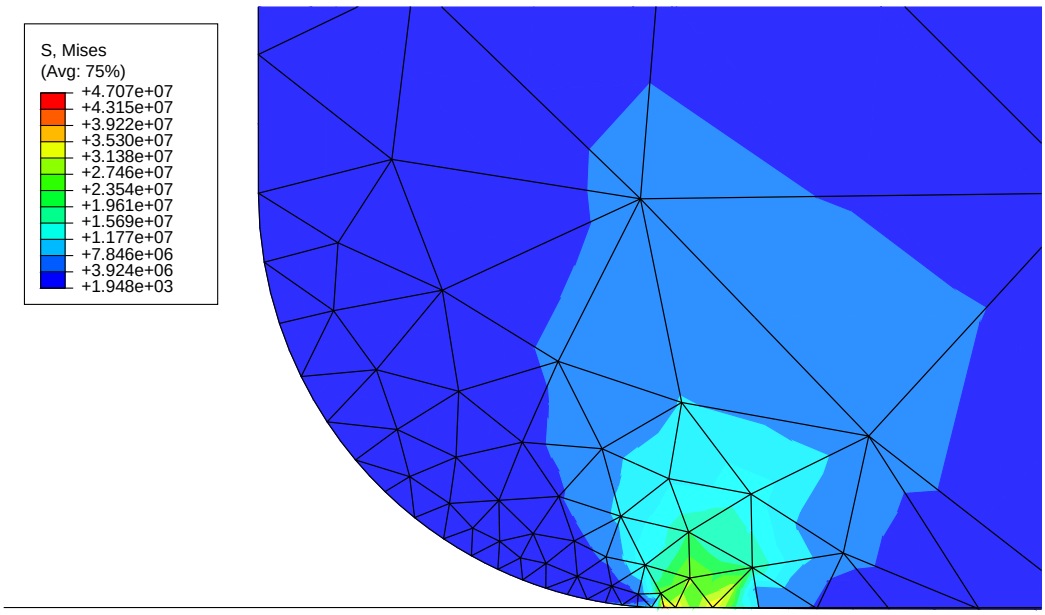


Figure 3.12: A close view of the rounded corner of the pin

with some radius, even if extremely small. To address both of these issues, an attempt was made to add a radius to the lower corners of the pin.

A radius of 6×10^{-6} m was applied to both the leading and trailing corners of the pin. The radius on the trailing edge had little effect on the stresses because the frictional shear on the bottom surface caused some separation on the trailing edge. At the leading edge, the radius of the corner was successively refined in order to reduce the angle that the element formed with the rail. It was found that decreasing the element size around the corner did not eliminate, nor did it even reduce the appearance of a spike in the stress at the transition from contact to non-contact at the leading corner. A plot of the stress in the pin in the region of the leading edge is shown in Figure 3.12. The red visible only in the elements at the transition from flat to rounded indicates that the stress at that point is much higher than the

surrounding region. Additionally, refining the mesh around the corner increased the time required to complete the analysis. Therefore, it was concluded that a squared pin was a reasonable modeling assumption.

3.6 Wear Calculation

The wear height (h) at each node is calculated according to the equation

$$\mathrm{h} = \frac{k_A P s}{H} \tag{3.2}$$

Which is derived by solving equation 2.6 for the wear volume and dividing by the contact area to get the height. The wear heights for each node are added to those from the previous step to create a total wear height, which is written to an output file at the end of the script. The total wear heights for the nodes can be averaged to produce an average wear height, or a line can be fit through them to find an average angle of the wear profile.

3.7 Pressure Distribution Sensitivity

This method is very sensitive to high pressure points in the pressure distribution. If there are nodes where the pressure is much higher than at the surrounding nodes, there will be an extremely large amount of wear at that node compared to the surrounding nodes. High pressure points can occur when the friction on the contacting surface causes the trailing edge of the pin to lift off of the disk. The portion of the

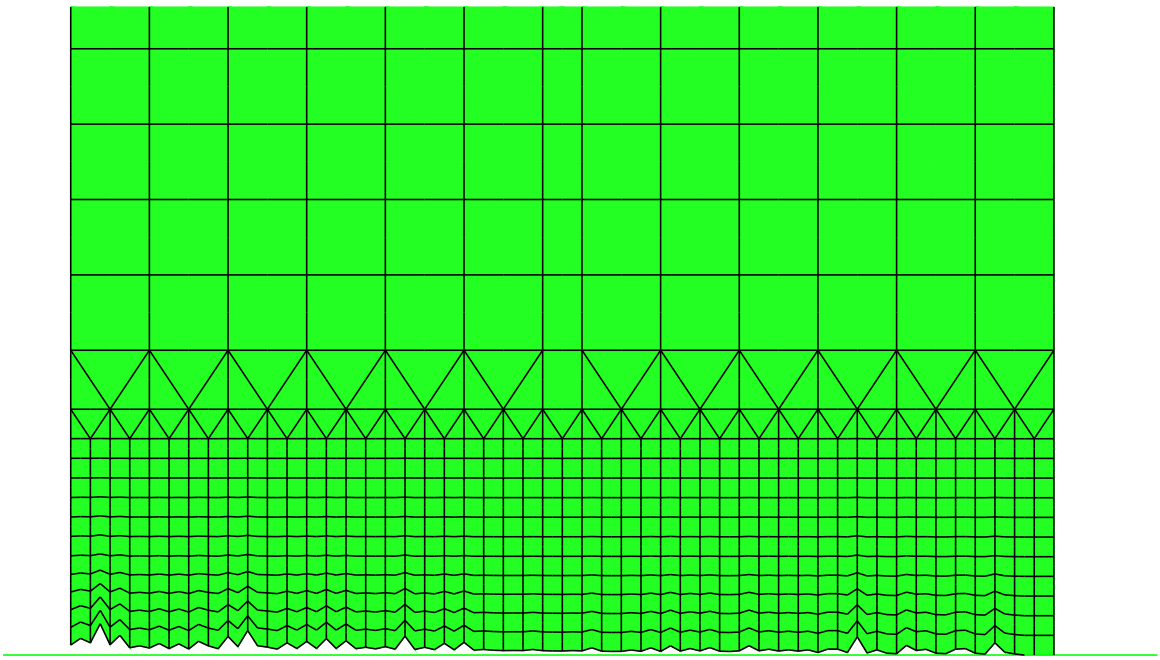


Figure 3.13: An example of an unrealistic wear profile that can cause an aborted analysis

disk not in contact does not wear, and a corner may form at the point where the surface changes from contact to no contact. This corner may become a point of high pressure in the next step. Once this sort of high pressure point is created, it often leads to more such points in the subsequent steps. This can result in elements with extreme amounts of distortion in the undeformed mesh. Figure 3.13 shows the type of element geometry that this can lead to. Some of the elements along the bottom of the pin have gone from being square to being chevron shaped. Not only is this sort of wear profile unrealistic, it may cause the analysis to abort. Therefore, it is important that either the creation of these high pressure points or their effects be eliminated or, at least reduced.

One way that their effects can be reduced is by using smaller distances for each step. This reduces the amount of wear for each step, so that even if there are high pressure points, they do not wear much, allowing the geometry to flatten. However, it also increases the number of steps needed for the pin to travel a given distance, resulting in a longer run time. Lowering the coefficient of friction or Young's modulus can also help eliminate high pressure points, but changing either of these results in a less realistic simulation. It is good, however, to remember that if switching to a harder material or one with a higher coefficient of friction there will be more of a problem with high pressure points. A third possibility is to spread the pressure to a few of the surrounding nodes. This method does not eliminate the high pressure points, but merely reduces the possibility that they will create excessively distorted elements. What happens in this case is that rather than having a single node with an unreasonable amount of wear, there is a small region with much greater wear than the rest of the surface. This may not completely eliminate high pressure points, but if the pressure is spread over enough nodes it will prevent an aborted analysis due to excessively distorted elements. It has been found that a five-node running average is sufficient to prevent most cases of excessive distortion due to high pressure points. A fourth option is to soften the contact algorithm. This is similar to the pressure averaging scheme, in that it allows the contact to be spread over a larger area, but it also allows the pin to penetrate into the disk. Benabdallah and Olender used a combination of changing the distance for each step and adjusting the contact

stiffness to create an acceptable solution. Here, a combination of pressure averaging and shortening the step distance is used.

3.8 Incremental Method Script

The script is written in Python because that is the language Abaqus uses for scripting. Using a script in Abaqus allows the kernel of Abaqus/CAE to be accessed without using the Abaqus/CAE gui. This allows tasks that are repeated many times to be automated. A script can be run either with or without the Abaqus/CAE gui. There are a few capabilities of the gui that are not available to scripts that are run without it.

While it was attempted to create the script to be as independent of the model as possible, there are still several things that are required of the model for the script to function. The most stringent requirement is that the model must be completely ready to analyze before the script is run. The second major requirement is that the mesh in the wear region be composed of rectangular elements in straight columns. This requirement is due to the nature of the mesh modification scheme. Also, the sliding direction must be parallel to the x-axis. Finally the script has names for many parts of the model hard-coded into it, either the model must conform to the names used in the script, or the names in the script must be changed to match the model. Other cases where either the script or the model is dependent on the other will be noted when covered in the walkthrough.

3.8.1 Versions

There are several versions of the script, each subsequent version included some additional feature or capability. The first version of the script included all the basic steps of running a job through Abaqus/CAE gui, extracting the pressure and calculating the wear, and modifying the model. Version 2 cleaned up the code and added the pressure averaging. Version 2.1 adds the capability to submit jobs to the command line. Version 2.2 adds the ability to increase the velocity in each step. Version 3 adds the ability to map the final state of stress from one step onto the beginning of the analysis for the next step. This version was not fully developed, and therefore not described in detail.

3.8.2 Script Version 2.1 Walkthrough

The heart of the script is a loop. Each iteration through the loop is referred to as a step. For each step an analysis is run, the needed data is extracted from the output, the wear for the step is calculated, and the model is modified. The pin moves a predefined distance each step. The loop is controlled by the distance traveled; when the total distance traveled reaches the maximum distance, the loop is exited. After the loop has been exited, the total wear at each node along the wearing surface is written to the output file, and the script is ended. Figure 3.14 is a simplified flow chart of the script. Each part of the script is explained in greater detail below.

The script begins with initializing constants and variables to be used throughout the rest of the script. Important constants include: the name of the model database

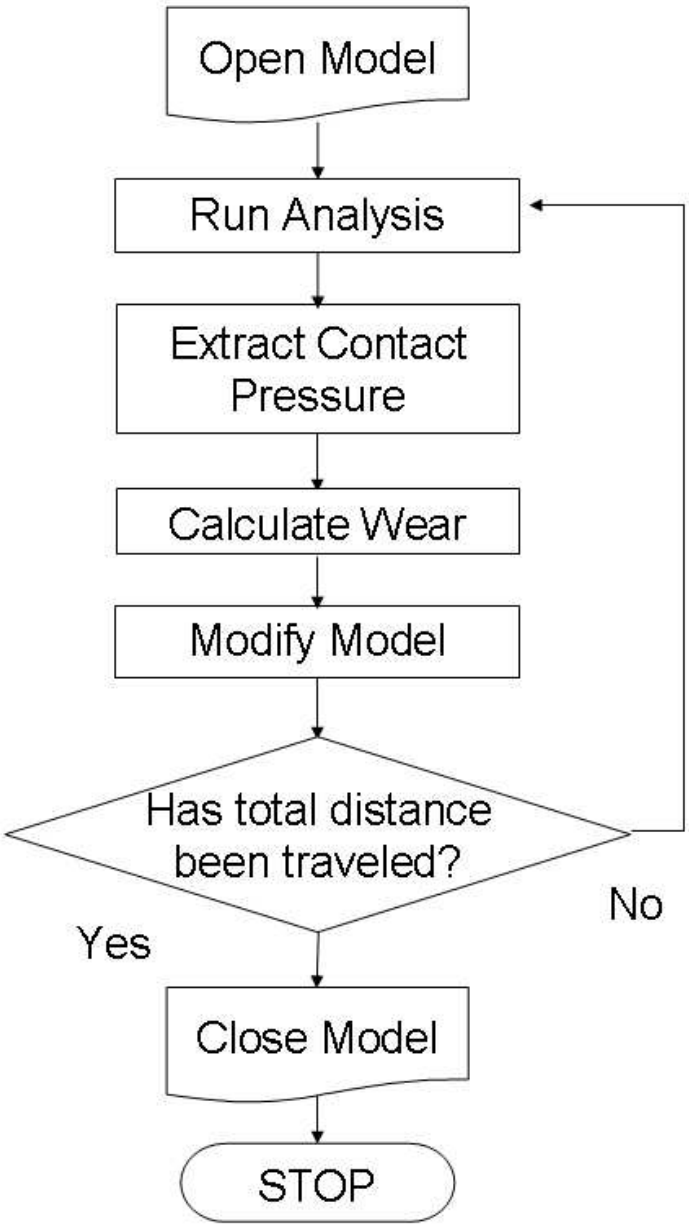


Figure 3.14: A flow chart outlining the basic steps of the script used in the incremental method of calculating wear

and the directory where it is located, the names of important parts of the model, the total distance to be traveled, the wear coefficient, and the hardness of the softer material of the two in contact. Variables that may be changed within the script include: the distance to be traveled in each step, velocity of the pin, the amount the velocity increases each step, and counters to control the loop.

Each step begins with running a structural analysis of the model. There are two methods to submit an analysis, called a job, to the solver. This is due to differences in the way that Abaqus/CAE kernel handles job submission depending on whether or not the gui is used. The preferred method runs the job through the Abaqus/CAE kernel, since this method is simpler and allows for better error handling. The other method submits the job to the system by spawning off another process, and has worse error detecting capabilities. If an error is detected, the loop will terminate. The second method should only be used if the first doesn't work, which was found to be the case when running the script without the Abaqus/CAE gui.

If a job is run through the kernel, the process is quite simple. A job is created in the model and submitted for analysis. When the job is completed, the status is checked to make sure that it completed successfully. If the job did not complete successfully, an error is returned.

If the job is run through the command line, the process is more complex. A job is created in the model and the input file is written. The job is run by spawning another process that runs the Abaqus solver with a system command. The job is

monitored for completion by checking for the existence of particular files created by the solver. If the appropriate combination of files is not found within a specified number of checks, an error is returned. This error checking process does not check for errors encountered during the analysis, only errors in the preprocessing stage, and errors causing the analysis to run for an excessive time. Because of this deficiency, an error in an analysis may not be caught until the next step, when it will likely cause more serious errors that will be caught.

After the analysis has been completed, the output database is opened and the contact pressure is read for each of the nodes on the contact surface of the pin. The pressure at each node is then averaged with the pressures at the two nodes to the left, and the two node to the right. This running average smooths the pressure profile to reduce the possibility of extreme distortion of elements due to high pressure points. Using the averaged pressure, the wear at each node is calculated according to equation (3.2).

Once the wear has been calculated, the model is modified to reflect the amount that the surface has worn. Like in Benabdallah and Olender's process, the interior of the mesh is adjusted as well as the surface. The wear region must have rectangular elements for this to function properly, because it depends on the nodes being in ordered rows and columns. For each node on the surface, the interior nodes along a line in the positive y direction and within a small deviation of x value, not greater than half the distance between the nodes on the surface in the original mesh, are adjusted by a fraction of the wear at the surface. The fraction of the wear that each

node is adjusted is determined by its distance above the surface node, and decreases logarithmically in finite increments as the distance from the surface increases. These increments are sized so that there is one node of the initial mesh in each interval. The interval size should be changed if the refinement of the mesh changes.

Additionally, a boundary condition is set to bring the pin in to contact with the disk at the beginning of the next analysis. The distance that the pin is moved is determined by finding the node with the lowest total wear height and multiplying that height by 1.001. The extra 0.1 percent is to account for any possible roundoff that could prevent the pin from actually coming into contact with the disk.

At the end of each step the counters are incremented, and if the final distance has been reached the loop terminates. After the loop ends, the total wear at each surface node is written to the output file, which is then closed. The model database is also saved and closed, and the script is ended.

IV. Incremental Method Results and Discussion

4.1 Results

To test the incremental wear script a test was run in which the pin was slid for a total of 200 meters in 10 meter steps with a velocity of 0.1 m/s, an applied pressure of 5 MPa and a wear coefficient (k_A) of 3.53×10^{-3}. The surface profile at the end of the 200 m is shown in Figure 4.1, and figures of the wear profile at each step are shown in Appendix B. The waviness of the surface is due to a high pressure point that developed at the rear (right) of the pin and progressed forward with each step. This sort of profile is not realistic. The waviness predicted by the script can be minimized or eliminated either by changing some aspects of the model, or by reducing the distance traveled in each step.

Rather than develop a new model, a second test was run with 0.01 meter steps. The smaller step size made the time required to travel the entire 200 meters prohibitively large, so it was only run for a total distance of 1 meter. Despite the shorter total distance, the smaller incremental size required 5 times as many steps as the previous test. The final profile is shown in Figure 4.2. This profile is much smoother than the previous one. Figures showing the evolution of this profile at 0.1 meter increments are in Appendix C. The average wear height across the pin is 2.85×10^{-5} meters. This compares very well with predictions using Archard's equation over the same distance, which predict a wear height of 1.76×10^{-5} meters.

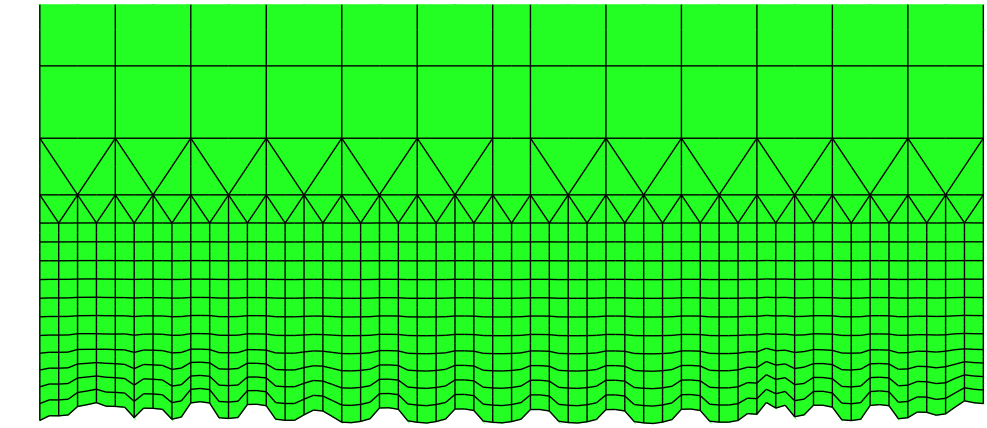


Figure 4.1: The profile of the pin predicted by the script after traveling 200 meters in 10 meter increments.

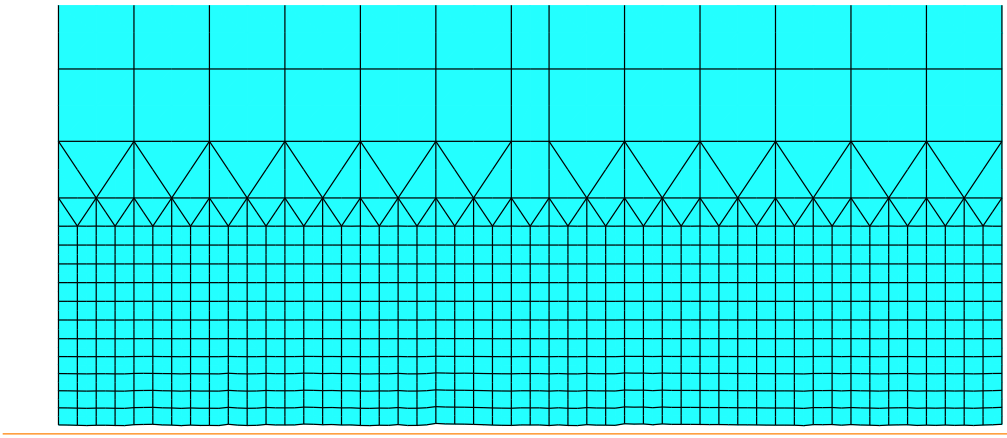


Figure 4.2: The profile of the pin predicted by the script after traveling 1 meter in 0.01 meter increments.

4.2 Discussion

The script used for the incremental method does a good job of matching wear height predictions to predictions from equations in the literature. However, it has not been verified against experimental results. If an accurate value of k_A is used, experimental results should be close to the predictions. The difficulty comes in finding an accurate value for k_A, as Podra and Andersson tell us: "the actual value of k_A for a particular contact should normally be experimentally determined" (20:73). Ideally values for k_A would be tabulated, but because wear is a system response, this is difficult.

The general form of this method has been well documented in the literature, and has been shown to be capable of producing accurate predictions. The implementation presented here has a few flaws that should be fixed before it is used in practice. These are mainly numerical issues that could take an extensive effort to iron out.

4.2.1 Need for Small Steps

The main issue is the small step size required to get a realistic profile. This severely limits the ability to use the script for events with larger distances, such as the wear of a slipper during a test at the HHSTT. The sensitivity of the script to high pressure points is the primary reason that small steps must be taken.

4.2.2 Pressure Averaging

The pressure averaging scheme implemented to combat these problems has its own problems. There is currently no method to ensure that the total reaction force on the pin from the disk remains unchanged when the pressure is averaged. This may result in a higher or lower pressure being input to the wear equation at each step.

4.2.3 Material History Effects

A lack of the ability to bring material history data from one step to the next is another problem with the script. There are two approaches to remedy this: (1) incorporate the wear steps within an analysis, or (2) map the results from the analysis in one step onto the model at the beginning of the next step.

To incorporate the wear calculations within an analysis is not possible to do in the context of the script presented here; The scripting capabilities in Abaqus only work with Abaqus/CAE, not the solvers. A user subroutine is required to integrate a process into a solver. Abaqus/Standard contains a user subroutine called UMESHMOTION that could be used to model the wearing of a surface, but it is not available in Abaqus/Explicit. While it was only tested with Abaqus/Standard, the script presented should work with both Abaqus/Standard and Abaqus/Explicit.

A method of mapping the solution from one analysis to the next was attempted, but was not fully developed. The method relies on a feature of Abaqus/Standard called *MAP SOLUTION. This functionality is intended for use with models that undergo large deformation that causes the mesh to be distorted such that it may no

longer provide accurate solutions. A new mesh is designed taking the deformed shape of the previous one, and the results are mapped to the new mesh. The inclusion of this feature required several changes to the model and the way that the script interacts with it. While the solution mapping ability is believed to be feasible to implement, it would restrict the use of the script to Abaqus/Standard. The user subroutine UMESHMOTION is recommended, over solution mapping because it provides a more refined solution. If future versions of Abaqus include one of these two features in Abaqus/Explicit, that method would be more attractive because it could be applied to a wider range of problems.

4.2.4 Mesh Modification

Another area for improvement is the mesh modification procedure. The current method does not take into account the geometry of the elements. Also, it has been conjectured that problems may arise when the distance between nodes becomes less than the discretization of the intervals in the algorithm. This could cause some elements to be resized while others are not. It is not known how this would affect the process. Additionally, The current method requires a rectangular mesh, which may not be suitable to some non-rectangular geometries. It is unknown what form this improved mesh adjustment algorithm might take, but it may be possible to use one of the adaptive remeshing techniques available in Abaqus.

V. Development of A Material Property Method for Calculating Wear

This chapter changes the focus from studying a method on a pin-on-disk model, to a slipper on rail model based on the HHSTT set up. In the material property method, wear is determined by the damage caused by the slipper running over an asperity. Unlike in the work of Cameron, who used a similar method, this method uses a microscopic asperity. To capture the activity on this scale requires elements on the microscopic scale. However, meshing the entire slipper with such small elements would create an intractable problem. Therefore, submodeling is used to limit the size of the model with such fine meshing. One submodel, called a local model, really only predicts the wear where the submodel is located. To determine the wear of the entire slipper, several local models can be taken from various locations along the bottom of the slipper. The wear calculated from each local model can then be used to approximate the distribution of wear rate along the entire slipper for a particular velocity and pressure. This can be done for several points throughout a test run down the HHSTT to determine the wear for the entire test. The goal of the current research is to determine the factors that are important in such a global-local analysis of a HHSTT slipper, and how to go about calculating wear from it. This chapter describes the development of the model and analysis. The next chapter will present an analysis at a single location on the slipper as a proof of concept.

While Cameron's numerical method for calculating wear is similar in the manner by which wear is found, the process presented here is really quite different. The primary difference is that the current method uses a finite element code, Abaqus, where Cameron used a hydrocode, CTH. Abaqus was chosen for several advantages it has over CTH. Abaqus can handle numbers down to 1×10^{-6} and further, which allows for modeling at the microscopic level. Also, in CTH the analysis time was limited to 10 μs, but with Abaqus analyses lasting several seconds or more are possible. Finally, Abaqus is more accessible and user friendly than CTH. Nearly all other differences stem from this.

5.1 Dynamic Analysis

This method uses a dynamic analysis in Abaqus/Explicit. Abaqus/Explicit was used because it tracks stress waves as they travel through an object, providing greater insight to the behavior of the material. There are also more material models available in Abaqus/Explicit. The damage models developed in section 2.5 are only available in Abaqus/Explicit, as is the Johnson-Cook plasticity model. Because Abaqus/Explicit models the stress waves, it is important that the transient effects caused by the loading of the slipper are damped out before it collides with the asperity. If the slipper hasn't reached a steady state of stress, it may be difficult to separate the effects of running over the asperity from the effects of the loading.

Only a short time is needed for the slipper to travel over an asperity, so initially a short time was used for the simulation and the loads and boundary conditions were

applied instantaneously. However this loaded the slipper faster than the stress waves could propagate through the slipper, causing elements to deform excessively. Figure 5.1 shows the effect of instantaneously applying a 1 m/s velocity, 2.2×10^{-6} seconds after the initial loading. The light blue part is the original, undeformed shape, and the dark blue part is the deformed model. Only the first three columns of elements have been compressed, while the rest of the model has not deformed at all. Despite only needing a short time for the slipper to traverse an asperity, it was necessary to ramp the velocity and pressure, then let the stress reach a steady state. A 1 second smooth ramp from 0 was found to bring both the velocity and the pressure up in an even manner. Steady state was reached after another 2 seconds at constant levels, bringing the total simulation time to 3 seconds.

5.1.1 Scale

For this method it is important to look at the problem at the microscopic scale, because the causes of wear occur at this scale. At scales on the order of 10^{-5} to 10^{-6} m, the granularity of metals begins to become apparent (15). Because of the granularity the material is not isotropic. This research will continue to assume isotropic material, but if later research refines the mesh further, the anisotropic behavior should be investigated. There are also some numerical difficulties involved with working at these scales: the maximum time step is reduced in explicit analysis methods due to reduced element lengths, and Abaqus/CAE has difficulty with numbers less than $1x10^{-6}$. To reduce these effects, the model uses millimeters as the base unit of length.

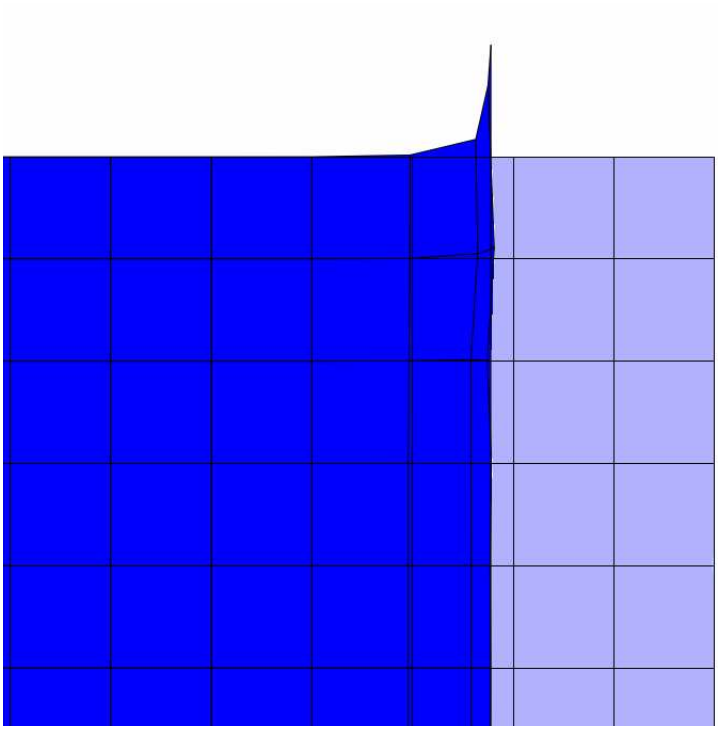


Figure 5.1: The right edge of the slipper crushed when a velocity condition was applied instantaneously.

This changes the magnitude of the numbers reported in Abaqus, but because all the inputs were scaled in does not change the results or the required time step.

5.1.2 Submodeling

Submodeling is another technique that is used to deal with the scale of the problem. Submodeling allows a small portion of a larger system to be modeled in detail while the rest of the system can be modeled coarsely. It achieves this by separating it into two analyses: a global analysis in which the whole system is analyzed, and a local analysis where the portion of interest is analyzed. The key to the technique is that the displacements from the global model are applied to the

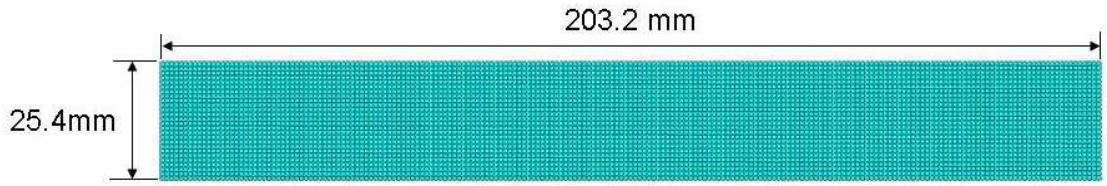


Figure 5.2: The global model of the slipper and its mesh

boundaries of the local model. The Abaqus Analysis Users Manual has a more detailed discussion of submodeling (9).

For this problem a plane strain model of a slipper from HHSTT running along a rigid rail is used as the global model. For the local model the rail is modeled as a deformable body, and a small, 1 mm, square is taken from the slipper as the submodel.

5.2 The Global Model

The global model is a plane strain representation of a slipper from the HHSTT. The slipper is 25.4 mm tall, 205.4 mm long and 101.6 mm deep. The model is meshed with rectangular elements, approximately 1 mm square, resulting in 5278 elements with 11,016 degrees of freedom. The dimensions of the slipper model, along with the mesh, are shown in figure 5.2.

A test was performed to determine the difference that a deformable rail would have on the stresses in the slipper compared to a rigid rail. Including a deformable rail in the global model caused the analysis to take a longer time to solve because of the added degrees of freedom in the rail. Also, the time required for the stresses to reach steady state was increased. The stresses did not achieve as large a magnitude with

the deformable rail model as they did with a rigid rail model. The rigid rail model was chosen because of the shorter analysis time associated with the fewer degrees of freedom. The rail is fixed in place and does not move.

The slipper has a uniform pressure of 15 MPa applied to the top surface. This pressure was chosen by looking at the peak slipper forces predicted by DADS. The entire slipper surface was assumed to be in contact. Averaging the maximum forces for each slipper location and dividing by the surface area of the slipper resulted in a pressure of 14.1 MPa. Using the maximum of all forces resulted in a pressure of 19.9 MPa. 15 MPa was chosen as a representative value that might be experienced by a slipper during a test. It should be noted that if only a portion of the slipper is in contact the pressure will be higher. There is currently no way to determine how much of the slipper is in contact at any given time. A velocity of 1 m/s was applied to the vertical faces of the slipper, excluding the corners. This location was chosen to avoid specifying the same degree of freedom twice, which could happen if the velocity was applied to the top surface, and to allow the bottom surface to deform in response to the contact. One meter per second is a low velocity for the HHSTT, but was chosen for two reasons: (1) it will allow for comparison with experimental results, and (2) it requires less time and distance to reach steady state.

From the global model the nodal displacements are written to the output database every 0.1 seconds. The output should be written using full precision to avoid potential roundoff errors. The displacements are interpolated between nodes and time steps when they are applied to the local model.

5.3 The Local Model

5.3.1 Slipper

The slipper submodel is a square with 1 mm sides and is shown in figure 5.3. It has submodel boundary conditions applied to the trailing edge, the top surface, and the top ninety percent of the leading edge. The submodel boundary condition is not applied to the bottom ten percent of the leading edge to allow it to deform when colliding with the asperity. The location of the local model is what determines what part of the global model drives the submodel boundary conditions. For the proof of concept the local model was taken just behind the leading edge of the global slipper model. The red lines in figure 5.3 indicate where the submodel boundary conditions are applied to the slipper in the local model.

The bottom ten percent of the slipper is meshed with 0.015 mm square elements. The rest of the pin is a mixture of triangular and quadrilateral elements that gets progressively coarser as the distance from the bottom increases. Only the bottom portion will develop enough stress and strain to be of interest for calculating wear, so it needs to be more refined while the rest of the pin can be meshed more coarsely to reduce the number of degrees of freedom. Figure 5.3 shows the mesh used on the local model of the slipper. There are 851 elements, with a total of 1800 degrees of freedom in the local slipper model.

The local model is where the outputs used to calculate wear and wear rate are determined. Therefore, care must be taken to ensure that data is written at the times

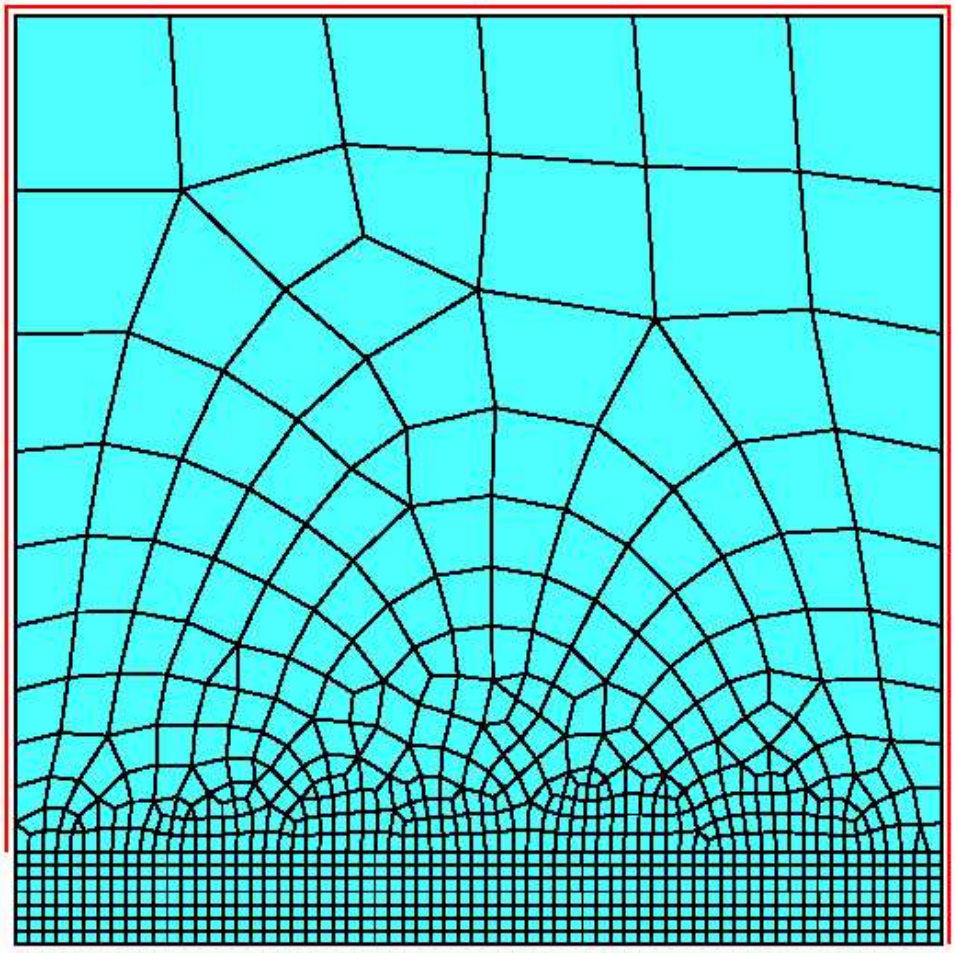


Figure 5.3: The mesh and boundary conditions for the local model, the red lines indicate the boundary condition location

it is needed for analysis. Output is only needed in a small time frame around the impact with the asperity. A list of time points to write the output can be defined so that the needed data is written without excessive amounts of unnecessary data also being written. Writing output at increments of 0.0001 seconds for the last 0.0015 seconds was found to be sufficient to provide the desired data.

5.3.2 Rail

In the local model, the rail is modeled as deformable, with a series of asperities at one end. The rail is clamped along the entire bottom edge. The rail is 0.6 mm tall with asperities that are 0.04 mm tall and 0.4 mm long, measured from peak to peak. The asperities are centered along the top edge of the rail, so the hight of the asperity above the flat section of the rail is 0.02 mm. Unlike in Cinnamon and Cameron, the asperities are circular arcs, not triangular (5; 6). Figure 5.4 shows the asperities and the mesh on them. There are eight elements along the surface of each half wavelength. The mesh becomes coarser as it goes to the right, beyond what is shown in the figure. The rail has 2029 elements, approximately one fourth of which are in the end of the rail with asperities, with 6606 degrees of freedom total.

In addition to being used to generate wear in the slipper, the asperities also provide some of the resistive force of friction. Therefore, the friction coefficient should be reduced along the asperity so that the total resistive force on the slipper does not change. Determination of the amount that the friction should be reduced to account for an asperity requires an additional study to correlate asperity size with frictional

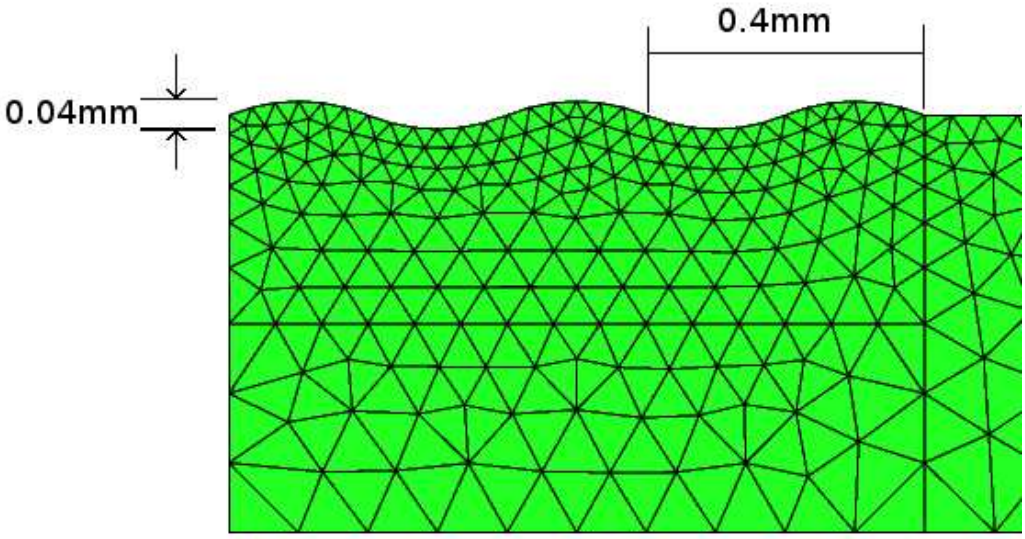


Figure 5.4: The end of the rail with asperities, showing the mesh and dimensions of the asperities

force. Because this data is not currently available, the friction coefficient was not changed in the region of the asperity.

This research was focused only on the development of the process; therefore, it was decided that an accurate value for the friction coefficient was not necessary. For both the global and local models a constant friction coefficient of 0.2 was used. This was chosen as an approximate value for the slipper and rail pair. An accurate value for the materials used would be required for wear prediction.

Initially the local model was to contain conformal asperities on both the rail and slipper, such as is shown in figure 5.5. When it became apparent that a longer time was needed to get the slipper to steady state conditions, the local model was changed

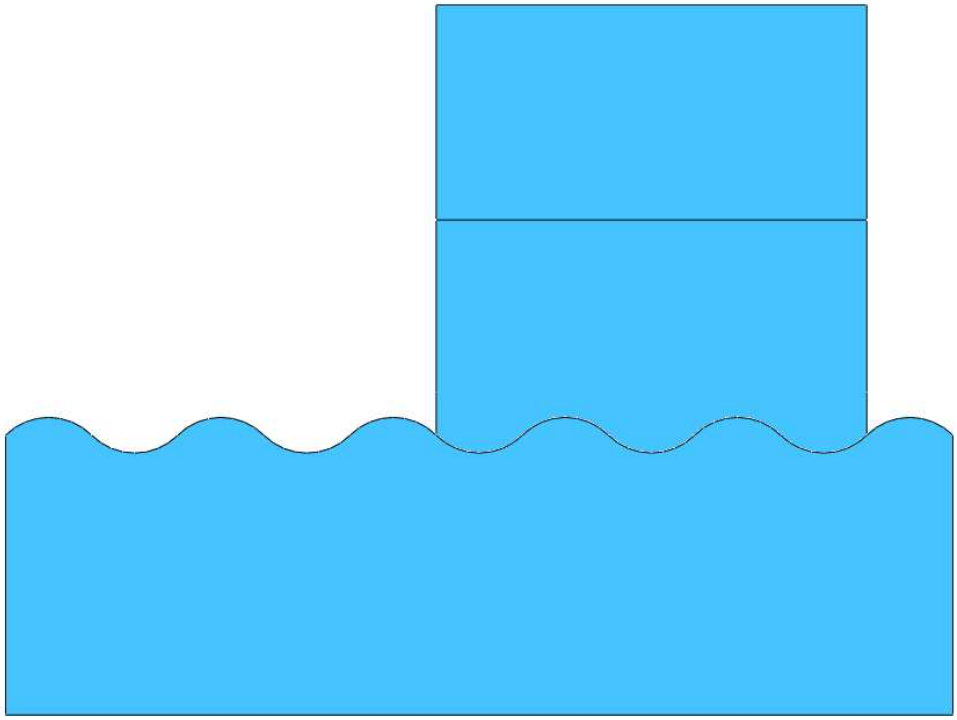


Figure 5.5: A preliminary version of the local model which was abandoned due to the need for more time to reach steady state.

to one with a flat slipper running over a flat rail until colliding with a deformable asperity at the end of the simulation.

5.3.3 Computing Cost Issues

The large increase in travel distance required a significant increase in the length of the rail in the local model. The total number of degrees of freedom for the whole model was 8406. Despite using the minimum length required for the rail, the computational cost was still significantly more than was acceptable. Therefore methods to reduce the computing cost of the local model were investigated.

The first thing tried was to break the rail into two parts: an analytical rigid part for the flat section of the rail, and a deformable part for the section with the asperities. Treating the majority of the rail as a rigid body reduced the degrees of freedom needed to solve the problem by approximately seventy-five percent. It was acceptable for the majority of the rail to be rigid, because its main purpose was to supply the contact condition while the slipper reached steady state. A deformable asperity section was desired because it would affect the reaction of the slipper. However, the slipper snagged at the transition from the rigid rail to the deformable rail and caused the analysis to abort before completing. An example of this snagging is shown in figure 5.6. It was decided to model the rail as a single deformable part to avoid this problem. A second attempt consisted of constraining all the degrees of freedom of the flat part of the rail, effectively making it rigid. This method did not reduce the computational cost. Further investigation of this method is not recommended.

The third method was to scale the mass of the rail to increase the size of the time steps. Mass scaling is an artificial increase of density to increase the maximum time step (Δt) allowable for stability. Initially, the mass was scaled for the entire rail. This method was successful in reducing the time required for the analysis to run. However, scaling the mass was found to cause the elements with scaled mass to act rigidly. Therefore, the section of the rail with the asperities was excluded from the mass scaling so that the asperities would deform when the slipper runs over them. This did not significantly increase the computing cost.

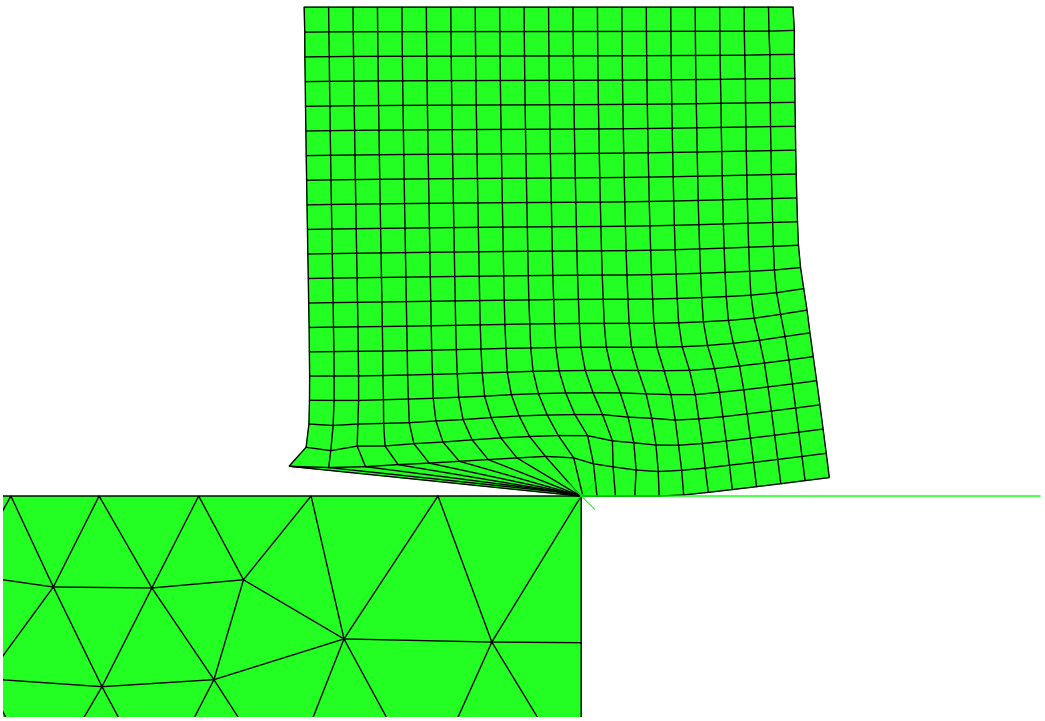


Figure 5.6: The slipper snagged on the transition from rigid disk (right) to deformable disk (left).

5.4 Material Properties

The slipper has the material properties of Vascomax 300, and the rail has the material properties of 1080 steel as determined by Cinnamon. The Johnson-Cook plasticity model was used. According to Cinnamon:

> The Johnson-Cook model relates the material flow stress (dynamic yield strength), σ, as:
>
> $$\sigma = [\mathrm{A} + \mathrm{B}\varepsilon^n][1 + \mathrm{C}\ \ln(\dot{\varepsilon}^*][1 - T^{*m}]$$
>
> where ε is the equivalent plastic strain, $\dot{\varepsilon}^*$ is the dimensionless strain rate ($\dot{\varepsilon}^*$=$\dot{\varepsilon}/\dot{\varepsilon}_0$) and T^* is the homologous temperature defined as:
>
> $$T^* = (T - T_{room})/(T_{melt} - T_{room})$$

Table 5.1: Material properties for 1080 Steel and Vascomax 300 (6:5-11)

Property	Vascomax 300	1080 Steel
E (GPa)	180.7	202.8
ν	0.283	0.27
ρ (kg/m^3)	8000	7800
JC: A (GPa)	2.17	0.525
JC: B (GPa)	0.124	3.59
JC: C (GPa)	0.0046	0.029
JC: n	0.3737	0.6677
JC: m	0.95	0.7525

where T, T_{room}, T_{melt} are the material temperature, room temperature, and material melt temperature, respectively (6:3-10). A, B, C, m, and n are material properties determined by experiment. The properties for Vascomax 300 and 1080 steel are shown in table 5.1. Additionally the damage initiation model described in section 2.5 was used for the calculation of wear. No published data could be found for either material. In lieu of data for the actual materials, damage data from example 2.1.3 in the Abaqus Example Problems Manual was used (10).

VI. Material Property Method Results and Discussion

6.1 Proof of Concept Results

A single global-local analysis was run as a proof of concept to show the feasibility of this method. The final global and local model configurations described in the previous chapter were used. The global model was run on four parallel processors and took approximately 0.5 hours to complete. The local model was also run on four parallel processors and took approximately 29 hours to complete. The deformed shape of the local model after running over one asperity is shown in Figure 6.1. Contours of shear failure risk parameter and ductile failure risk parameter are shown in Figure 6.2, and Figure 6.3, respectively. The data is plotted on the deformed geometry in part (a) of each figure, and the same data is plotted on the undeformed geometry in part (b). For each figure, the distance traveled over the asperity is 0.4 mm. Neither of these figures shows any damage, which means that no wear is predicted.

The lack of wear is not surprising. The slipper only traveled 0.4mm over the asperity. The slipper would only wear a minuscule amount in this distance. It may be possible to predict wear if the slipper were run over a series of asperities, rather than just one. This would introduce a new problem due to the nature of the submodel boundary conditions on the local model. Because the global model does not account for any wear at the surface, the pin will not continue to be pressed against the rail with the same force as it deforms and moves down the rail in the local model. This

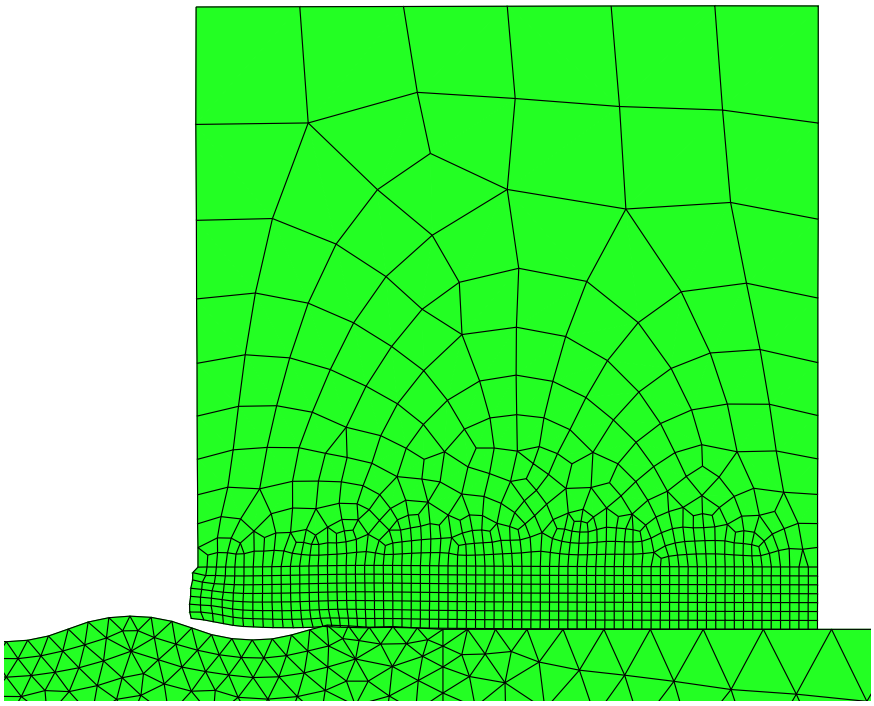
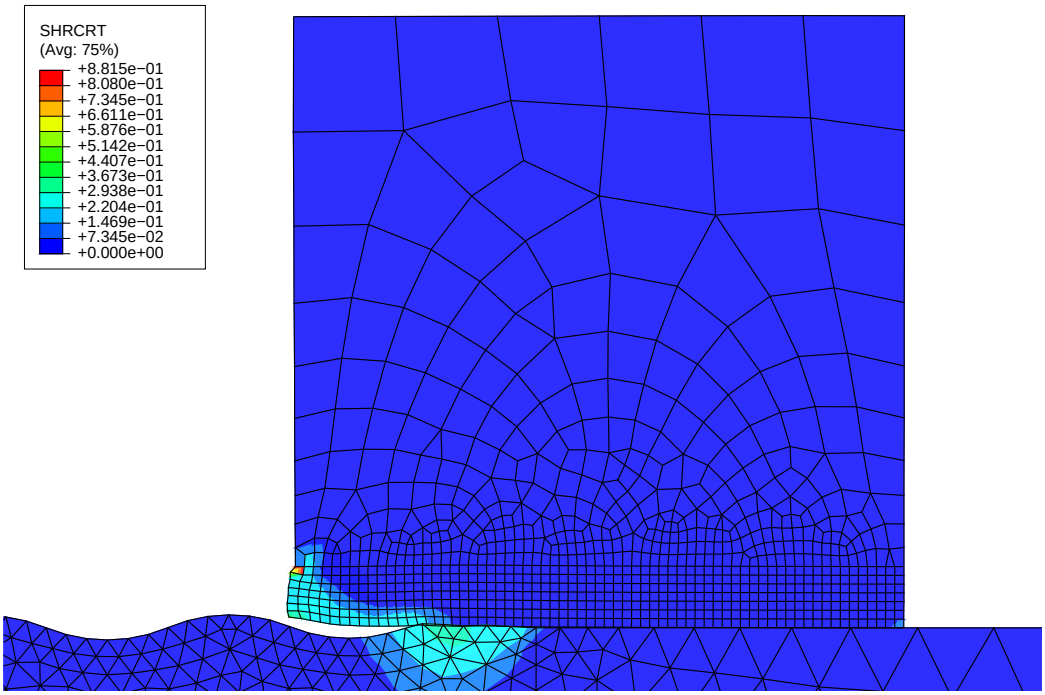


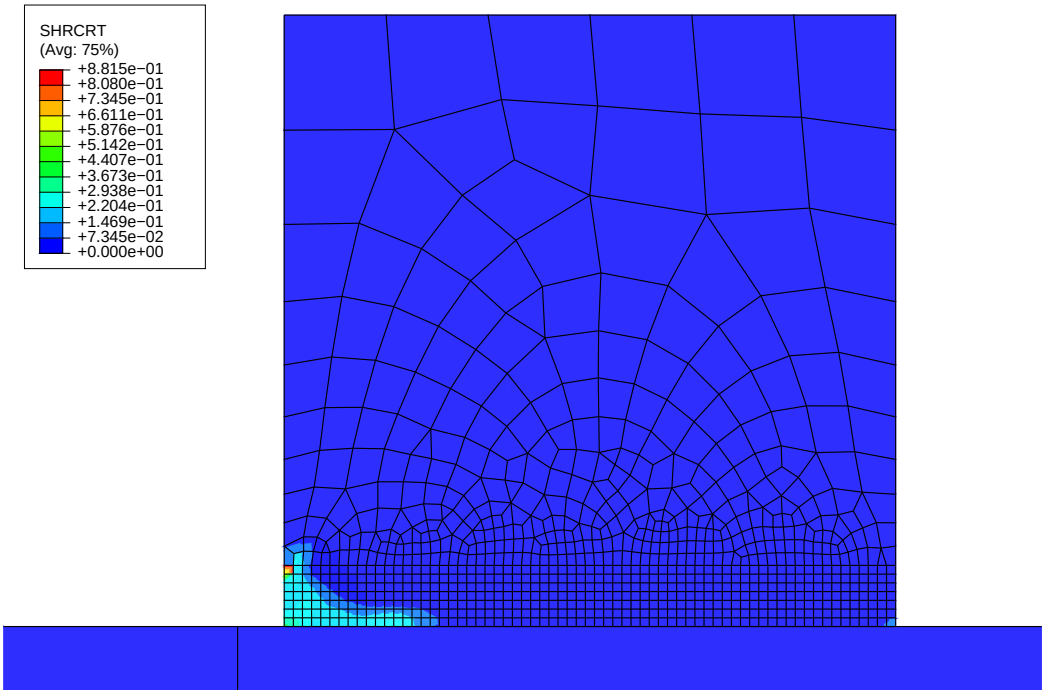
Figure 6.1: The deformed geometry of the local model after running over a single asperity

may not be a problem for small travel distances with small amounts of wear, but it may become a larger issue as the wear in the local model increases.

If the results from the case where the asperities were rigid due to mass scaling are considered, there is damage. The deformed geometry is shown in Figure 6.4. While the asperities did not deform, the contact formulation does allow for the surfaces to penetrate each other a small amount. This analysis could be representative of either a case with a harder rail, or a softer slipper. Figures 6.5 and 6.6 show contours for the shear and ductile failure risk parameter, respectively, again on deformed geometry in part (a) and on undeformed geometry in part (b).

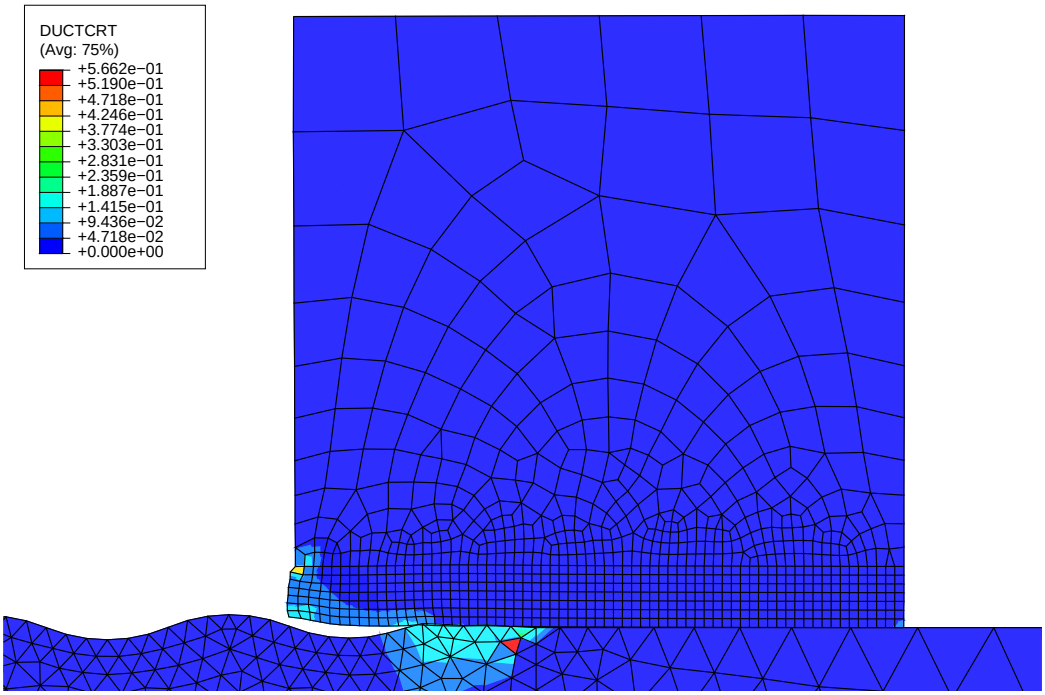


(a) deformed geometry

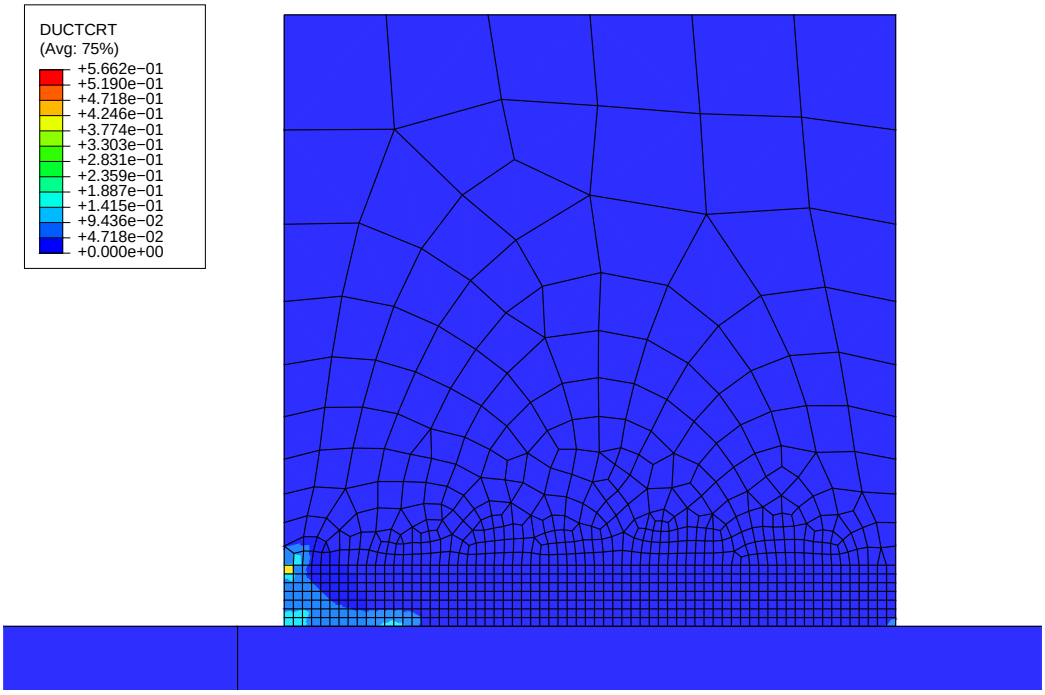


(b) undeformed geometry

Figure 6.2: Contours of the shear damage criterion after running over a single asperity



(a) deformed geometry



(b) undeformed geometry

Figure 6.3: Contours of the ductile damage criterion after running over a single asperity

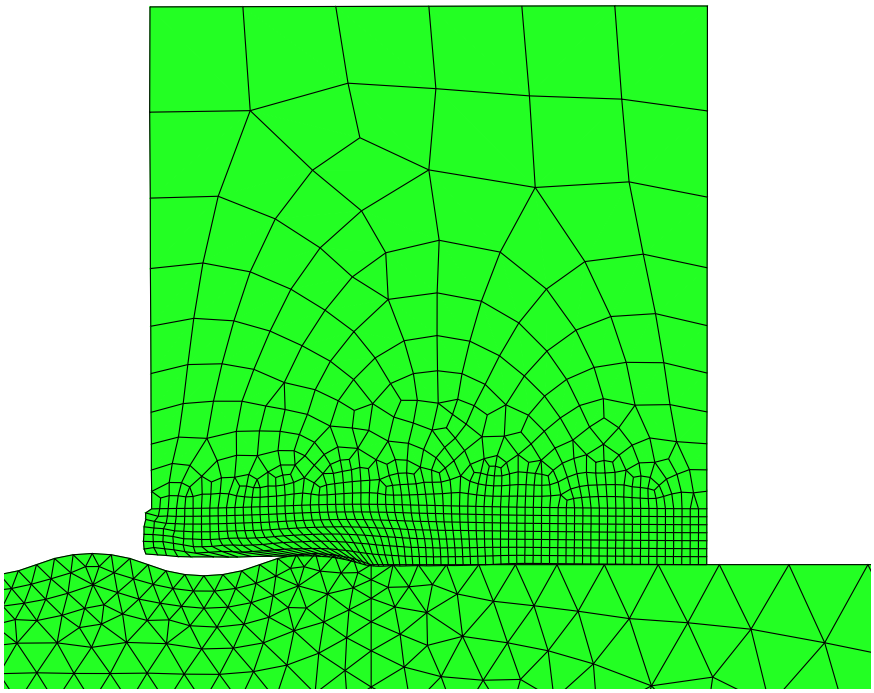
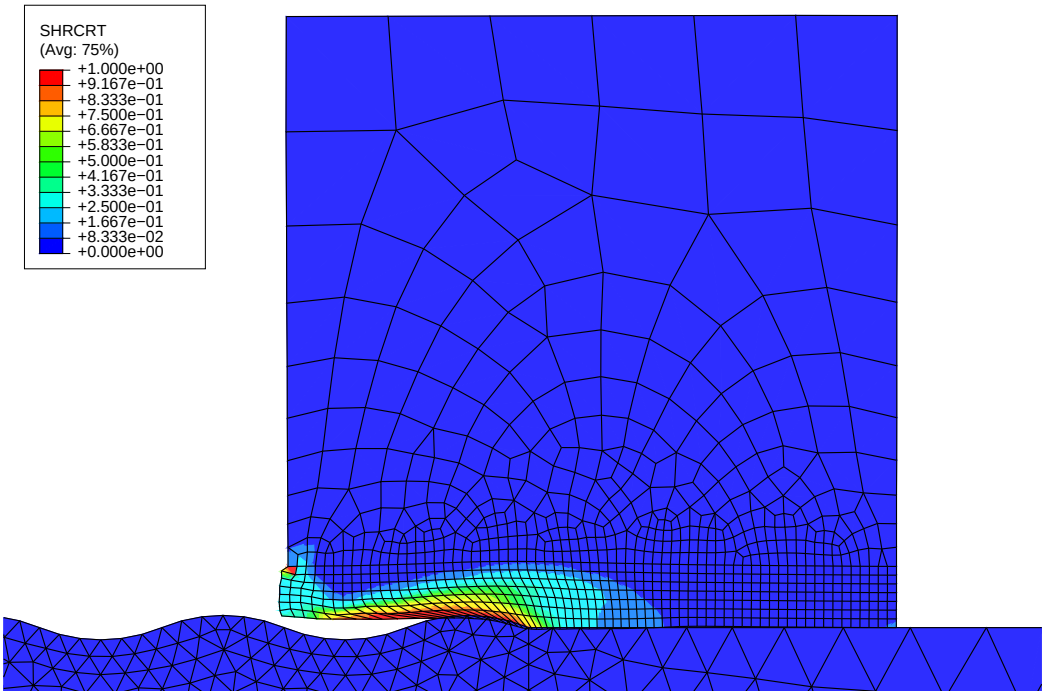


Figure 6.4: Deformed geometry for the case with rigid asperities after running over a single asperity

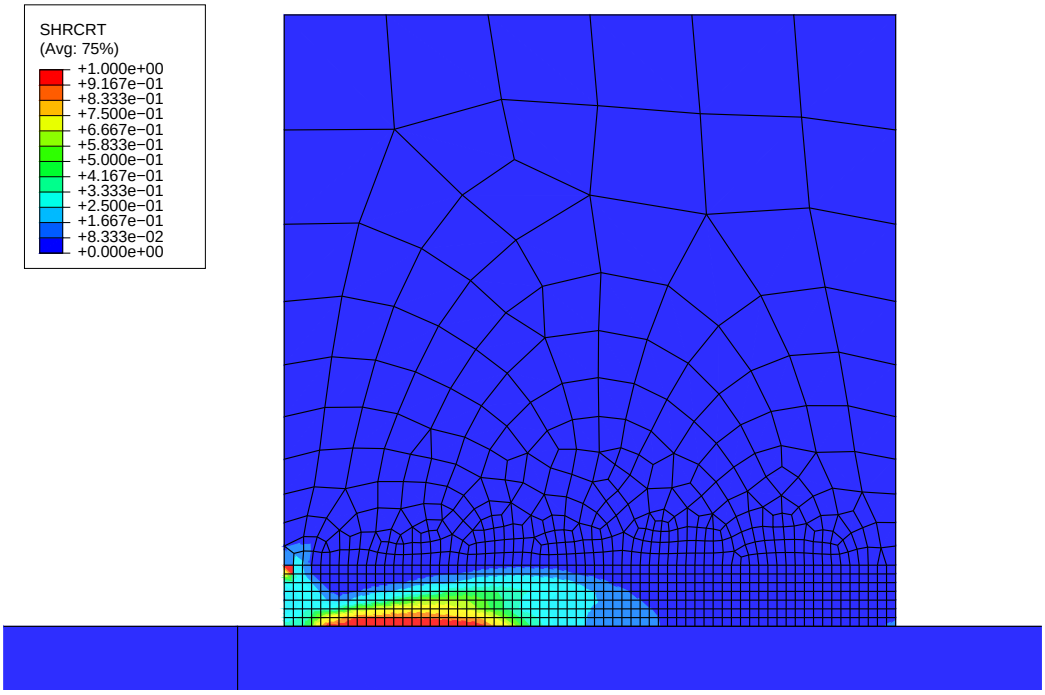
In Figure 6.5 (b) regions with values of 1 or higher are colored red, and are damaged. There are approximately 13 elements that are damaged, for an area of 2.925×10^{-3} mm^2. Multiplying the area by the depth of the model (101.6 mm) gives a wear volume of 0.29718 mm^3. This corresponds to a dimensional wear rate of 0.74295 mm^3/mm for a distance of 0.4 mm. By normalizing by the area of the slipper, 20,645 mm^2, a non-dimensional wear rate of 3.5987×10^{-5} is obtained. The ductile damage criterion does not predict any damage.

6.2 Discussion

According to Lim and Ashby, the non-dimensional wear rate should be between 10^{-3} and 10^{-9} for mechanical wear. The non-dimensional wear rate predicted by the

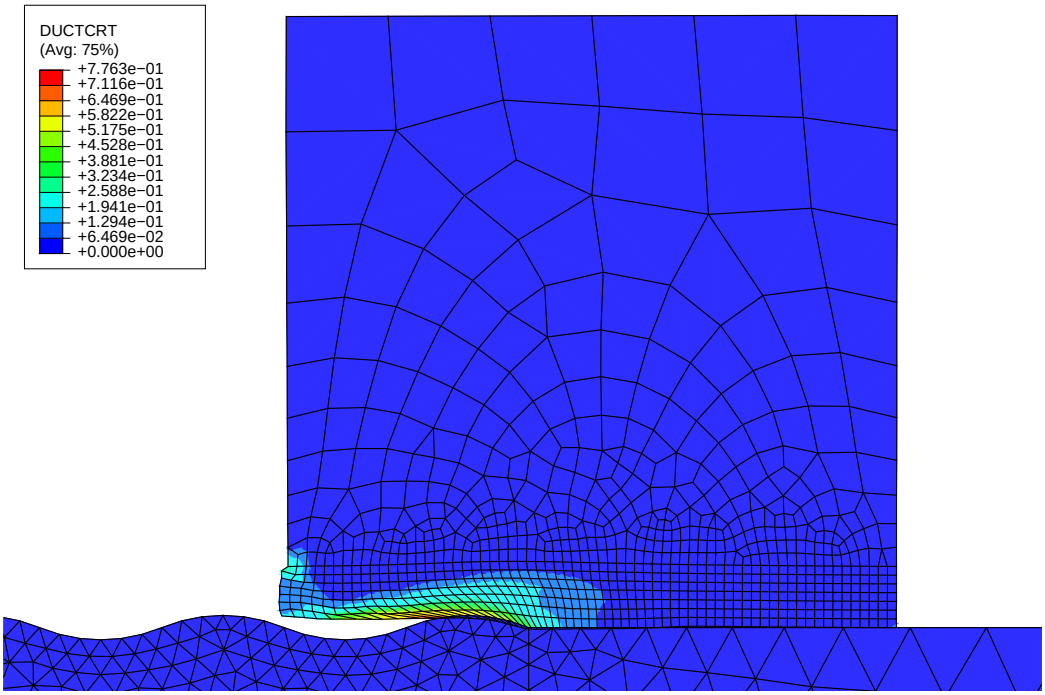


(a) deformed geometry

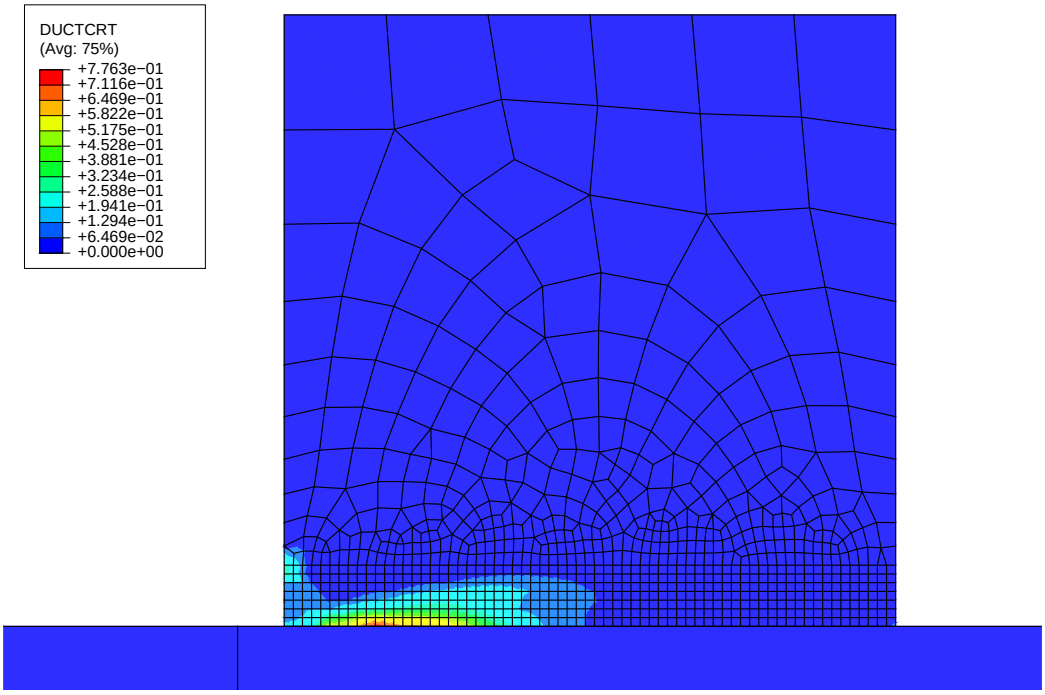


(b) undeformed geometry

Figure 6.5: Contours of the shear damage criterion for the case with rigid asperities after running over a single asperity



(a) deformed geometry



(b) undeformed geometry

Figure 6.6: Contours of the ductile damage criterion for the case with rigid asperities after running over a single asperity

material property method is in the middle of this range. This shows that the process can be used to predict experimental wear rates. The accuracy of the predictions will depend on the correct asperity size, and an accurate damage criteria being used.

A wear rate for the entire slipper could be calculated by taking local models from several locations along the length of the slipper and calculating a wear rate for each. The wear rates would then be plotted against their position along the slipper, and a curve fit through them. The average wear rate would be the rate for the slipper as a whole. If a series of these tests were done for a variety of applied pressures and a velocities, the data could be tabulated for use in a macroscopic wear prediction method, such as the one presented in Chapter 3.

VII. Summary and Conclusions

Wear is a difficult phenomenon to predict. Slipper wear at the HHSTT is even harder due to the extremely high velocities of the sleds. Two different methods for predicting wear were investigated; a macro-scale method based on Archard's wear law, and a micro-scale method based on material properties. These methods were tested at low speeds so that comparisons could be made with results from the literature. Both were shown to be feasible methods of accurately predicting wear.

7.1 Incremental Method

7.1.1 Conclusions

The incremental method was implemented with a Python script for Abaqus. The ability to use scripts with Abaqus is very well suited for this problem. The integration with Abaqus/CAE eliminates the need to spend energy and time creating input files or parsing output files by automating the process.

The incremental method script will produce realistic values for wear. The accuracy will obviously depend on the correct parameters being input into the model and script. For the 1 meter test case, the wear predicted by the script was only 1 mm more than was predicted by Archard's wear law. This is not a surprising result, because similar methods have been shown in the literature to produce good results.

The primary problem encountered with the script is that it is very sensitive to numerical issues. Many numerical issues were solved in the development, but many more still remained. While these issues could be worked out, they were a major motivation for developing the material property method. In order to get reasonable wear profiles, very small steps had to be taken, and the pressure had to be averaged over several nodes. A reduction of the numerical issues should also reduce the need for these in the method.

The wear profile is closely related to the pressure distribution on the sliding surface. The friction due to sliding is the primary factor in determining the pressure distribution. Less friction will result in a flatter distribution, and more friction in a more steeply sloped distribution of pressure. If there is enough friction, the trailing edge of the pin will lift off of the disk.

The pressure distribution was not affected by the type of analysis carried out. This is because the pressure was only taken at one point for each analysis. With the current implementation, a dynamic analysis does not add much, if anything, to the accuracy of the prediction.

Smaller steps produce better results, but also require more computing time. The largest step size that gives accurate results should be used.

7.1.2 Areas for Future Improvement

There are several things that could be done to improve this implementation of the incremental method. If it is to be developed further, the following things should be improved upon.

The mesh modification algorithm presented is not very robust. A more general and capable solution should be developed. A new modification algorithm should allow for irregular meshes, and should be based on the current mesh at each step.

Several improvements could be realized by integrating the method within an analysis. Smaller steps could be taken with less increased cost, because the number of file reads and writes needed would be reduced. Additionally, material history would be carried through each wear calculation.

7.2 Material Property Method

7.2.1 Conclusions

The material property method was shown to be feasible as a method for predicting wear. The wear rate predicted was in the range of those predicted by Lim and Ashby. With the proper calibration, the accuracy would be improved. There are, however, several factors that influence the method's wear predicting abilities.

Before the slipper runs over the asperity there is no damage predicted. An asperity was required to generate wear. Also the ductile criterion did not predict any wear in either case presented.

To capture the micro-scale phenomena, a fine mesh must be used. However, there is some lower bound where a finite element analysis no longer accurately models the material behavior. This limits the element size that can be used.

It is important to ramp applied conditions slowly, because a wave propagation code is used and stress waves take time to propagate through the material. Additionally, the stress waves must be allowed to be damped to a steady state before the slipper runs over the asperity.

It was found that a deformable rail greatly increases the time required to compute a solution. Mass scaling can be used to decrease the time, but reduces the ability of the rail to deform. A rigid rail also increases the stresses in the slipper.

7.2.2 Areas for Future Study

Only a preliminary investigation of the primary factors important to the process was conducted. A more detailed development, calibration, and validation is required to apply the process to real situations.

The asperity size is directly related to the amount of wear. Therefore, it is important to use the appropriate asperity size for the problem. Currently there is no data prescribing the correct asperity size to use. A possible way to determine the appropriate asperity size is to look at friction. When the slipper collides with an asperity, a force is imparted to the slipper. The force that an asperity of a particular size imparts to the slipper can be tabulated as a function of the slipper pressure and velocity. This data could then be cross-referenced with the coefficient of friction data

to choose the asperity size that produces the same frictional force. Once this has been done, a frictionless contact, or lower coefficient of friction, should be specified on the asperities.

The criteria used to determine wear should also be studied to determine which is the most appropriate. Besides the two presented here, a third possibility is a maximum energy criterion based on the material's stress-strain curve. By integrating the under the stress-strain curve, the energy reached at failure would set an upper limit on the amount of energy that could be absorbed by the material. Any region where the maximum energy is exceeded would be considered worn. This criterion would not require additional experiments to determine damage parameters. Other criteria may be investigated as well.

Finally, the method needs to be expanded to capture wear for the entire slipper throughout a test. Several submodels should be taken at multiple locations along the length of the slipper to get a more accurate prediction of the way the entire slipper wears. Additionally, the global model should be run with various pressures and velocities to get predictions for an entire HHSTT test.

```
# Python script to calculate wear in a pin on disk system
# Created 2007, by Aaron Chmiel
#########################################################
# This section imports stuff from abaqus
import site
import math
import os
import part
import assembly
import step
import load
import mesh
import job
import meshEdit
from abaqusConstants import *
from abaqus import *
from odbAccess import *
#############################################################
# Initialize constants and variables

print 'Script began'
# Set the working directory, where the model database is located
PWD='I:/My Documents/Wear/'
# open the output file
output=open(PWD+'BigWear_output.txt','w')
# Open the model database
mdb=openMdb(pathName=PWD+'BigPin2.cae')
```

```
# Set the names for model parts
# Model Name
mod=mdb.models['Model-1']
# Root Assembly
assemb=mod.rootAssembly
# Name of Pin Instance
part1=assemb.instances['BigPin-1']
# Portion of the mesh to be adjusted
wearRegion=assemb.sets['Wear_Region'].nodes
# Name of the BC to move the pin into contact with the disk
BC=mod.boundaryConditions['Pin-Contact']
VBC=mod.boundaryConditions['Velocity']

# Set the important constants
max_distance=400    # Maximum total sliding distance, meters
dist=100     # Initial amount to step the total distance, meters
K=3.53e-3     # Wear coefficient
hardness=1e3     # Hardness of the pin
output.write('Total distance to be traveled: ' + str(max_distance)
      + ' millimeters \n' +  'Number of steps to take: '
      + str(max_distance/dist) + '\n\n')

############################################################
# Compare X
# This function compares the x coordinates of a list of nodes
# with each entry in the form:
#  [Node Label, Pressure, [x-coord, y-coord, z-coord], ...]
```

```
def Comp_X(x,y):
if x[2][0]>y[2][0]:
return 1
elif x[2][0]==y[2][0]:
return 0
else: # x[2][0]<y[2][0]
return -1

#############################################################
# Get average pressure
# This function extracts contact pressures from an abaqus
#output database and averages the pressure over 5 nodes
#
# Input:
#  odbname: the name of the output database to be opened
# Additional Requirements:
#  Comp_X: this function is used to sort the list for averaging
# Output:
#  avePress: a list of nodal pressures averaged over up to 5 nodes
#   each entry has the form:
# [Node Label, Pressure, [x-coord, y-coord, z-coord]]

def getPressAve(odbname):
  odb=openOdb(odbname)
  lastFrame=odb.steps['Step-1'].frames[-1]
  pressure=lastFrame.fieldOutputs['CPRESS']
  press=[[0,0]] # sets the first element to [0,0]
  for n in pressure.values:
gridPt=part1.nodes.getFromLabel(n.nodeLabel)
```

```
coord=assemb.getCoordinates(gridPt)
   press.append([n.nodeLabel,n.data,coord])
  press=avePress=press[1:] # removes the first element
  press.sort(Comp_X)
  print 'pressure extracted'
  index=0
  while index<len(press):
sum=0
tally=0
if index!=0:
  sum=sum+press[index-1][1]
  tally=tally+1
if index!=1:
  sum=sum+press[index-2][1]
  tally=tally+1
sum=sum+press[index][1]
tally=tally+1
if index<len(press)-1:
  sum=sum+press[index+1][1]
  tally=tally+1
if index<len(press)-2:
  sum=sum+press[index+2][1]
  tally=tally+1
average=sum/tally
avePress[index][1]=average
index=index+1
  odb.close()
  print 'pressure averaged'
  return avePress
```

```
#################################################################
# Lim & Ashby Wear
# This function calculates the wear at each node in a
# list given pressure distance traveled, a non-dimensional
# wear constant and matererial hardness.
#
# Input:
#  press: List of nodal pressures with each entry in the form:
# [Node Label, Pressure, ...]
# dist: Constant, distance traveled in for each analysis run
# Additional Requirements:
#  Wear Coefficient
# Material Hardness
#  Distance traveled
# Output:
#  List of nodal wear with each entry of the form:
# [Node Label, Wear]

def LimWear(press,dist):
# K is the wear coefficient
deltatime=1
distance=dist
wear=[[0,0]] # sets the first element to [0,0]
for p in press:
delwear=p[1]*K*distance/hardness
wear.append([p[0],delwear])
wear=wear[1:] # removes the first element
```

```
print 'Wear Calculated'
return wear

#################################################################
# comp_Y
# Compares Y coordinates of a list with each entry in the form:
#  [label, [x-coord, y-coord, z-coord], ...]

def comp_Y(x,y):
if x[1][1]>y[1][1]:
return 1
elif x[1][1]==y[1][1]:
return 0
else: # x[1][1]<y[1][1]
return -1

#################################################################
# modifyModel
# This function updates the model to reflect the results
# of the last analysis. It takes the wear calculated at
# the bottom of the pin, and adjusts the nodes interior
# to the pin so that the elements stay approximately square.
# Also enables a boundary condition to move the pin into
# contact with the disk.  And, it sets the results to map
# to the model.
#
# Inputs:
# wear: a list of incremental nodal wear, each entry
# must be of the form: [Node Label, Wear]
```

```
# minwear: A constant with the minimum amount of wear
# Additional Requirements:
#  Comp_y: this function is used to sort the list of
# nodes in a column by their y-coordinate

def modifyModel(wear,minwear):
  # Apply the wear
  for n in wear:
    # nodeObject for node along bottom of the pin
    gridpt=part1.nodes.getFromLabel(n[0])
      # gets the coordinates of the node
      coord=assemb.getCoordinates(gridpt)
      # sets the first entry to the bottom node
      nodesAbove=[[n[0],coord,gridpt]]
      for m in wearRegion:
        # if the node in wearRegion is above the node n
        if abs(m.coordinates[0]-coord[0])<0.0000075:
          # add it to the list
          nodesAbove.append([m.label,m.coordinates,m])
    nodesAbove=nodesAbove[1:]   # removes the first element
    # Sorts according to y coordinate (not really needed for now)
    nodesAbove.sort(comp_Y)
    for q in nodesAbove:
      if abs(q[1][1]-coord[1])<=1.5e-2:
assemb.editNode(q[2],offset2=n[1])
      elif 1.5e-2<abs(q[1][1]-coord[1])<=4.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.90)
      elif 4.5e-2<abs(q[1][1]-coord[1])<=7.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.80)
```

```
      elif 7.5e-2<abs(q[1][1]-coord[1])<=10.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.70)
      elif 10.5e-2<abs(q[1][1]-coord[1])<=13.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.60)
      elif 13.5e-2<abs(q[1][1]-coord[1])<=16.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.50)
      elif 16.5e-2<abs(q[1][1]-coord[1])<=19.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.40)
      elif 19.5e-2<abs(q[1][1]-coord[1])<=21.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.30)
      elif 21.5e-2<abs(q[1][1]-coord[1])<=24.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.20)
      elif 24.5e-2<abs(q[1][1]-coord[1])<=27.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.10)
      elif 27.5e-2<abs(q[1][1]-coord[1])<=30.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.09)
      elif 30.5e-2<abs(q[1][1]-coord[1])<=33.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.08)
      elif 33.5e-2<abs(q[1][1]-coord[1])<=36.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.07)
      elif 36.5e-2<abs(q[1][1]-coord[1])<=39.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.06)
      elif 39.5e-2<abs(q[1][1]-coord[1])<=41.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.05)
      elif 41.5e-2<abs(q[1][1]-coord[1])<=44.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.04)
      elif 44.5e-2<abs(q[1][1]-coord[1])<=47.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.03)
      elif 47.5e-2<abs(q[1][1]-coord[1])<=50.5e-2:
```

```
assemb.editNode(q[2],offset2=n[1]*0.02)
      elif 50.5e-2<abs(q[1][1]-coord[1])<=53.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.01)
      elif 53.5e-2<abs(q[1][1]-coord[1])<=56.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.009)
      elif 56.5e-2<abs(q[1][1]-coord[1])<=59.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.008)
      elif 59.5e-2<abs(q[1][1]-coord[1])<=61.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.007)
      elif 61.5e-2<abs(q[1][1]-coord[1])<=64.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.006)
      elif 64.5e-2<abs(q[1][1]-coord[1])<=67.5e-2:
assemb.editNode(q[2],offset2=n[1]*0.005)
      elif 67.5e-2<abs(q[1][1]-coord[1]):
assemb.editNode(q[2],offset2=n[1]*0.004)
  print 'Pin Worn: 1 increment'

  # Move the pin to the disk
  BC.resume()
  BC.setValues(u2=minwear*-1.001)

#################################################################
# an analysis error class

class AnalysisError(Exception):
value='error'
def __init__(self,value):
self.value=value
def __str__(self):
```

```
return repr(self.value)

#################################################################
# Run Job
# This function runs the analysis
#
# Input:   jobname

def runjob(jobname):
    step1=mdb.models['Model-1'].steps['Step-1']
    # Create a job
    mdb.Job(name=jobname,model='Model-1')
    # Run the job
    mdb.jobs[jobname].submit()
    output.write(str(jobname) + ' started \n')
    mdb.jobs[jobname].waitForCompletion()
    if str(mdb.jobs[jobname].status)=='COMPLETED':
      output.write(str(jobname) + ' has completed \n')
    else:
      raise AnalysisError, 'Analysis did not complete properly'

#################################################################
# Run Job 2
# This function runs an analysis on the command line
#
# Input:  jobname, oldjobname

def runJob2(jobname):
   jobs='job='+jobname
```

```
   mdb.Job(name=jobname,model='Model-1')
   mdb.jobs[jobname].writeInput()
   pid=os.spawnl(os.P_NOWAIT,'abaqus','abaqus',jobs)
   print pid
   output.write('PID:='+str(pid)+'\n')
   os.waitpid(pid,0)
   limit=0
   done=bool(False)
   while not done:
     if os.path.exists(jobname+'.log'):
       if limit>8e4:
         raise AnalysisError, 'Analysis did not complete properly'
         break
       if os.path.exists(jobname+'.odb'):
         if not os.path.exists(jobname+'.lck'):
            done=bool(True)
            print limit
       limit=limit+1
     else:
       raise AnalysisError, 'Analysis did not complete properly'

##############################################################
# Add Wear
# This function adds the incremental wear to the total wear to
# keep a running total wear at each node.
# The arguments can be in any order, it should work the same.

def addWear(sum,delta):
    if len(sum)!=len(delta):    # if they are not equal length
```

```
        sum=delta    # set them equal
    else: #if len(sum)==len(delta):  # if they are equal length
        for s in sum:    # loop over sum
            for d in delta:    # loop over delta
                if d[0]==s[0]:    # if the labels are the same
                    s[1]=s[1]+d[1]    # add the wear to the sum
    return sum

#################################################################
# Find Minimum of the second values of a list of lists

def minimum(superlist):
    list=[100]
    for w in superlist:
        list.append(w[1])
    list=list[1:]
    minimum=min(list)
    return minimum

#################################################################
# The Actual program
# This calls all the above functions and calculates accumulated
# wear over the total travel distance

i=1 # a counter for the number of steps
total_dist=0 # a counter for the total distance
wear=incwear=[[0,0]]
while total_dist<max_distance:
output.write('Begin wear step ' + str(i) + ' \n')
```

```
jobname='BigWear_step-' + str(i)
try:
  #        uncomment the version you want to use, and
  #        uncomment the other one
   runjob(jobname)
   #       runJob2(jobname)
except AnalysisError, error:
   print error
   break
pressure=getPressAve(jobname+'.odb')
incwear=LimWear(pressure,dist)
wear=addWear(wear,incwear)
minwear=minimum(wear)
modifyModel(incwear,minwear)
total_dist=total_dist+dist
output.write('End Wear step ' + str(i) + ' \n')
i=i+1
for n in wear:
  output.write('The total wear at node: %5d is %8.7e \n' % (n[0],n[1]))
mdb.save()
mdb.close()
output.close()
print 'Script ended'
```

Appendix B. Wear Progression Images for the First Test of the Incremental Method

This appendix contains images of the wear profile after each step of the script for the first case described in chapter IV. Only the bottom portion of the pin is shown to conserve space. The progression of the high pressure spot from the rear of the pin to the front is clear. Each image is after another 10m of travel. The pin is traveling to the left.

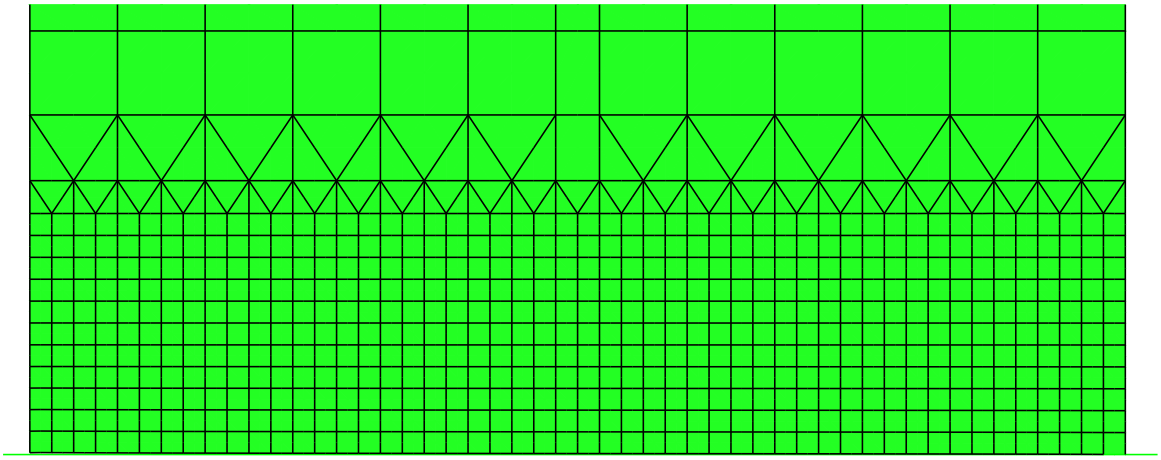


Figure B.1: Wear profile after 10m

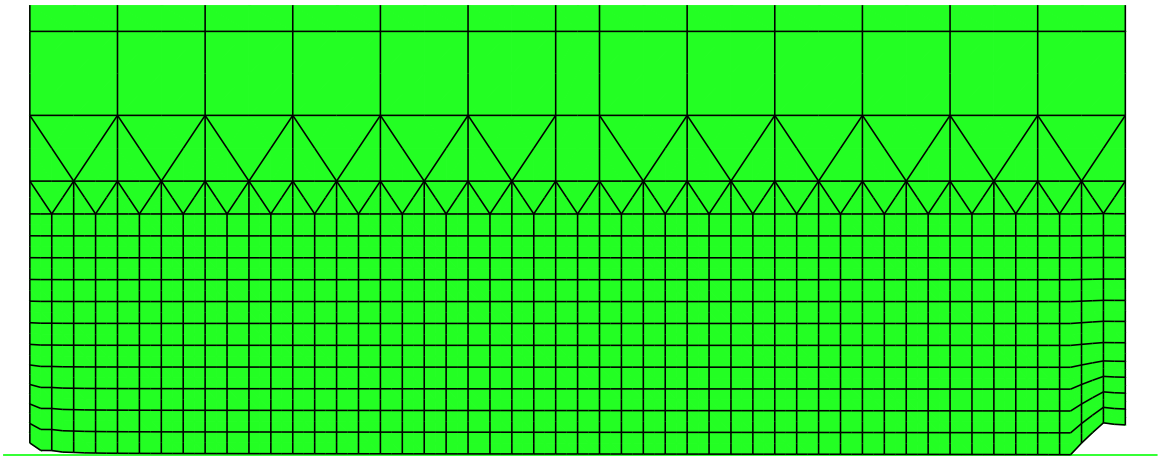


Figure B.2: Wear profile after 20m

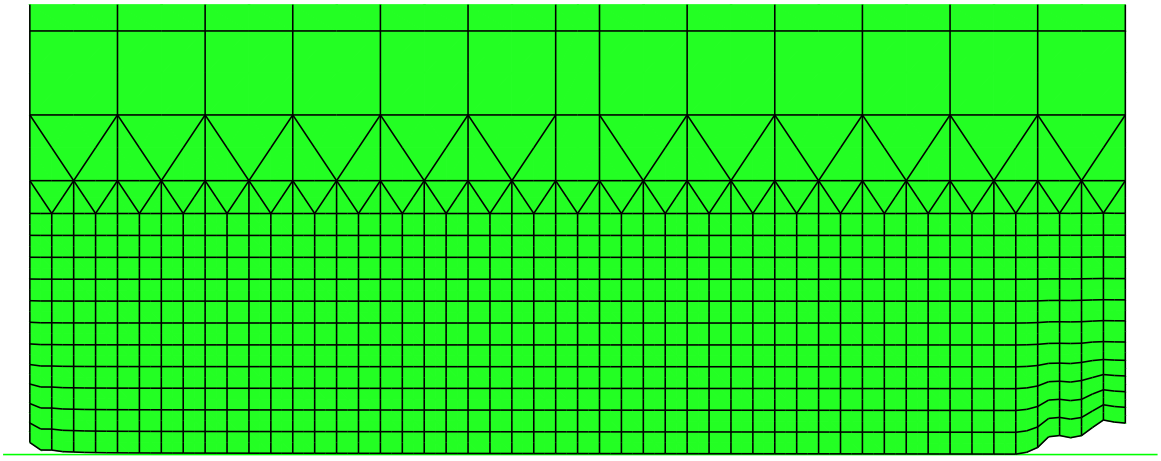


Figure B.3: Wear profile after 30m

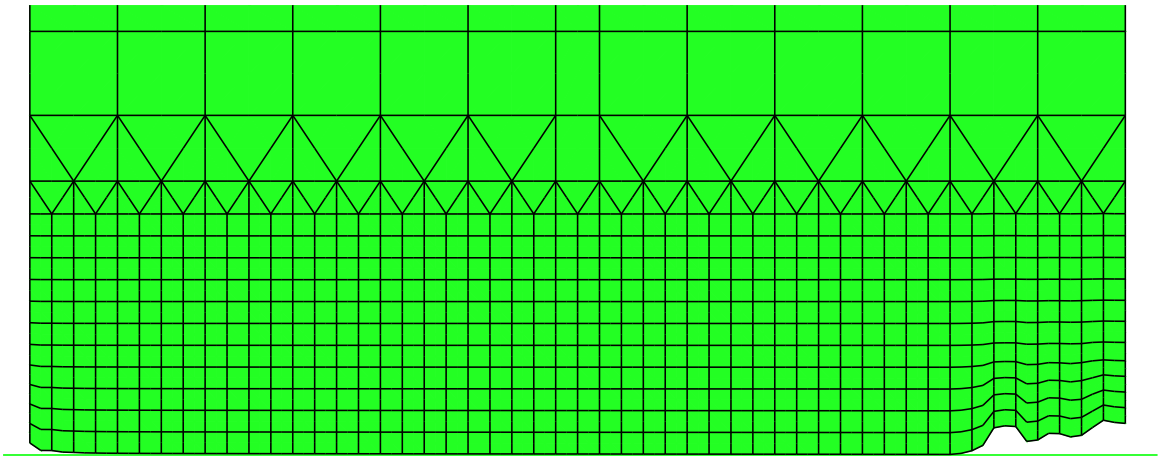


Figure B.4: Wear profile after 40m

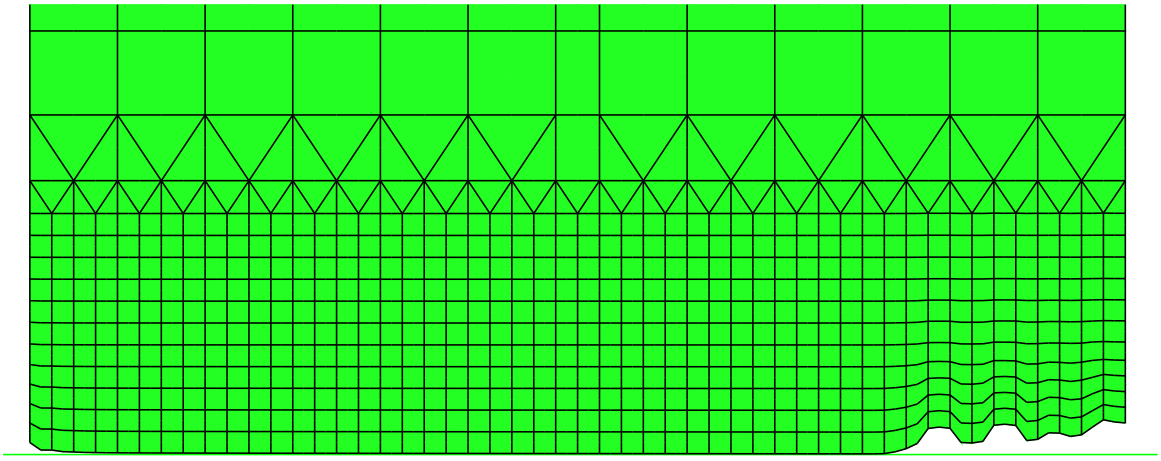


Figure B.5: Wear profile after 50m

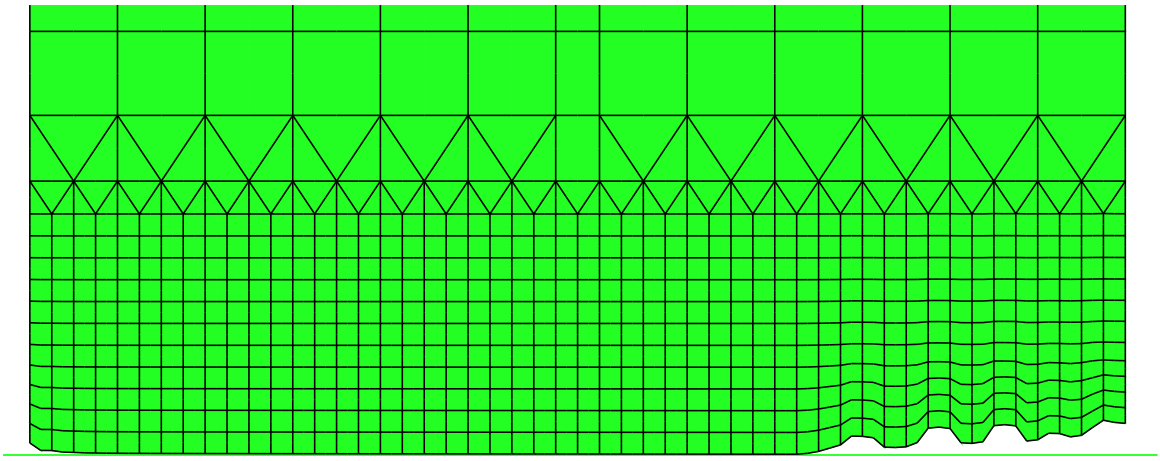


Figure B.6: Wear profile after 60m

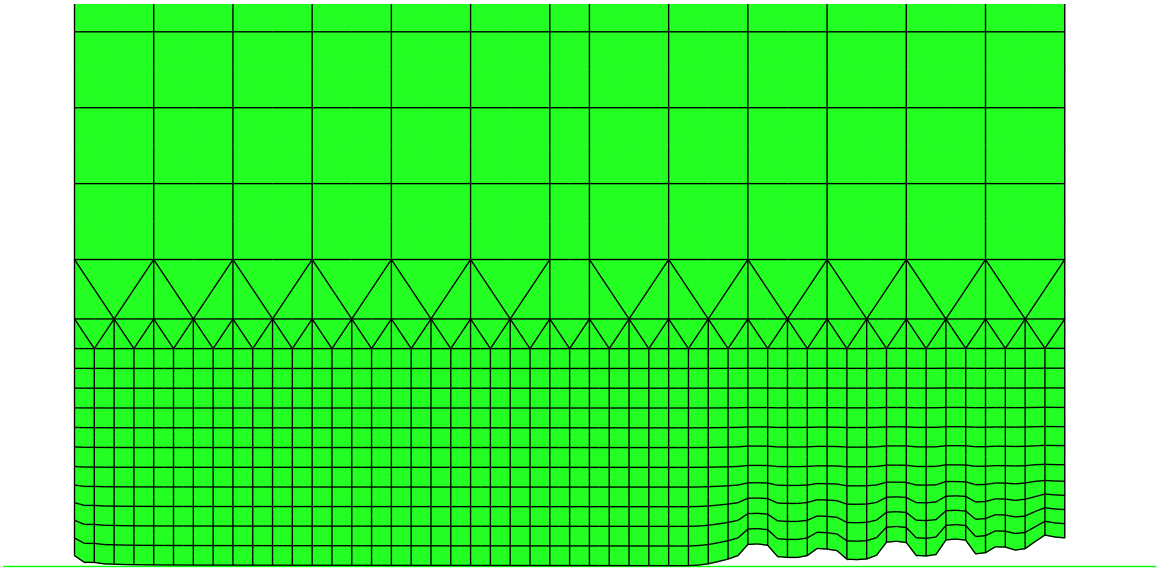


Figure B.7: Wear profile after 70m

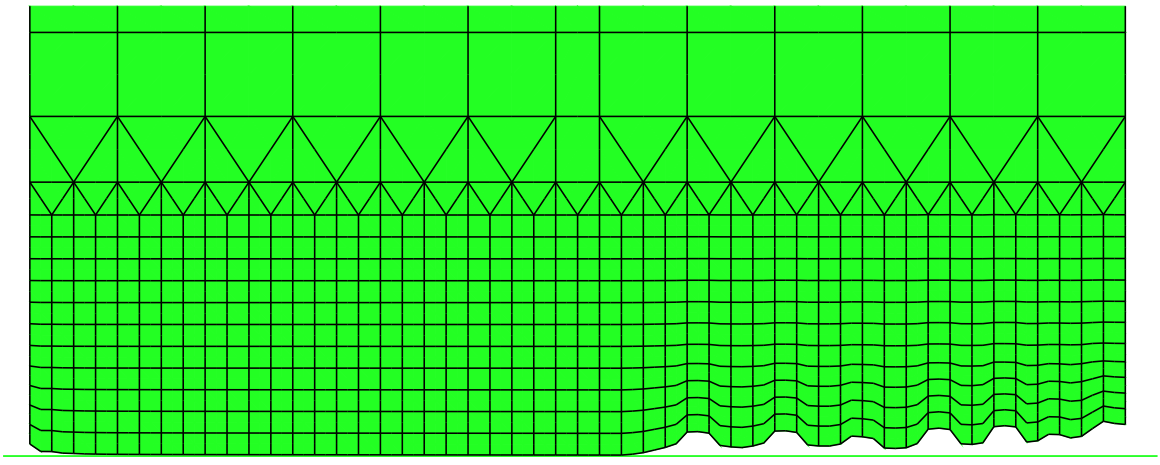


Figure B.8: Wear profile after 80m

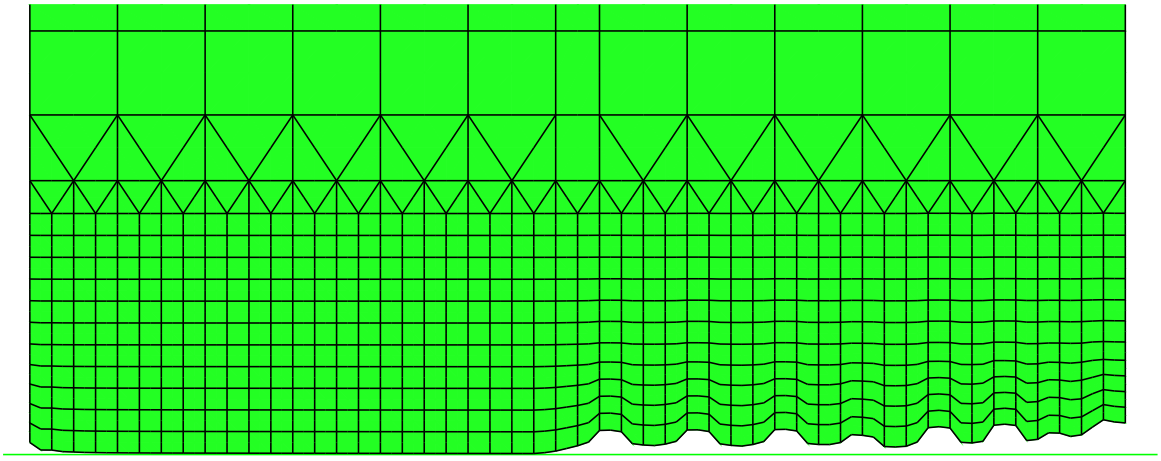


Figure B.9: Wear profile after 90m

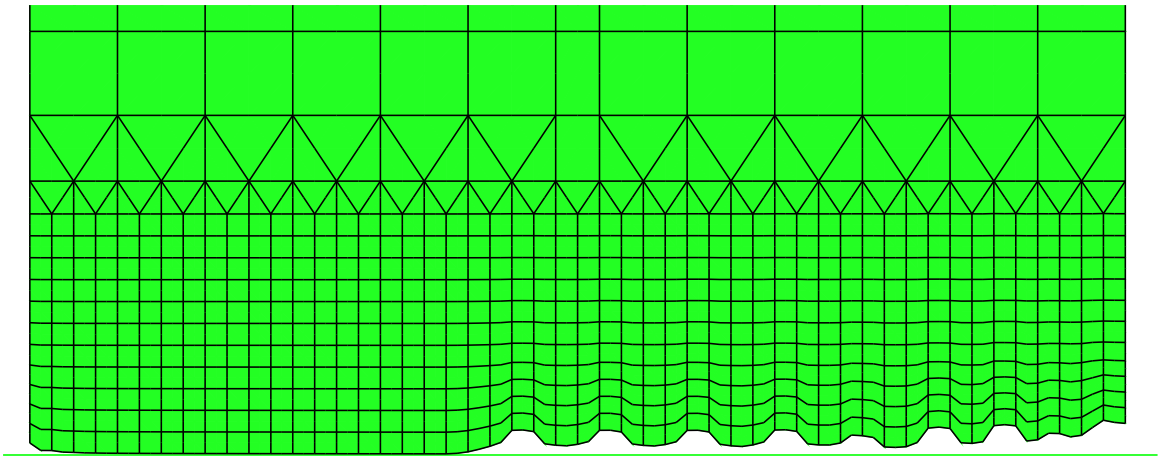


Figure B.10: Wear profile after 100m

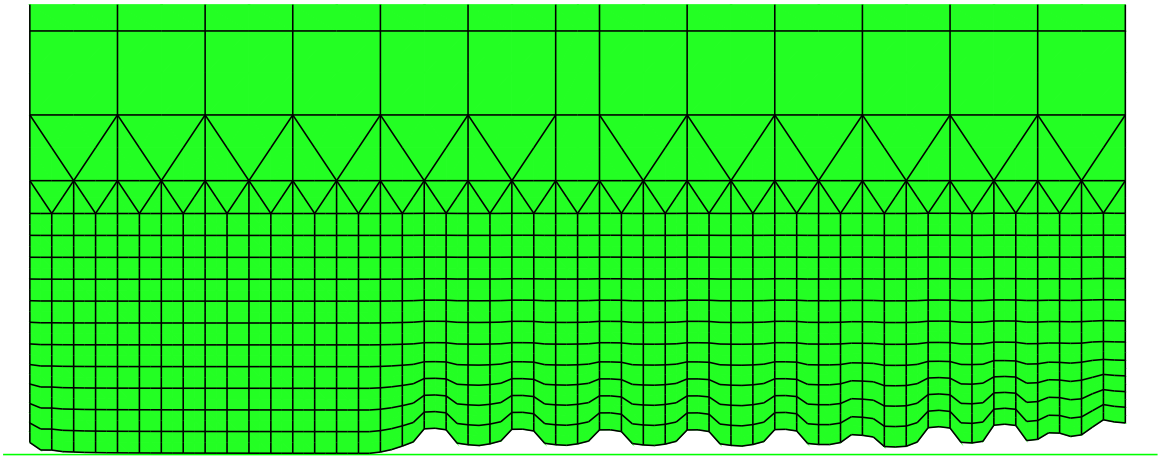


Figure B.11: Wear profile after 110m

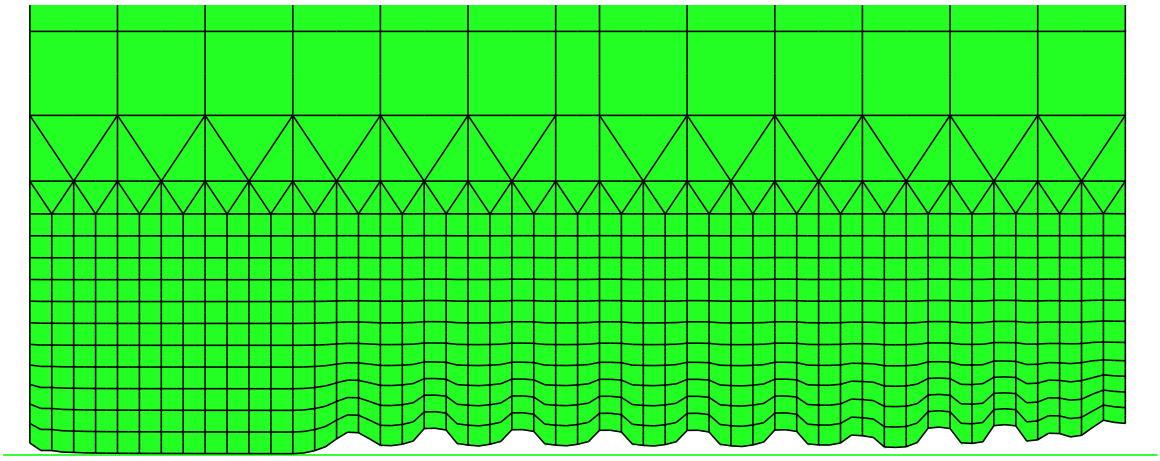


Figure B.12: Wear profile after 120m

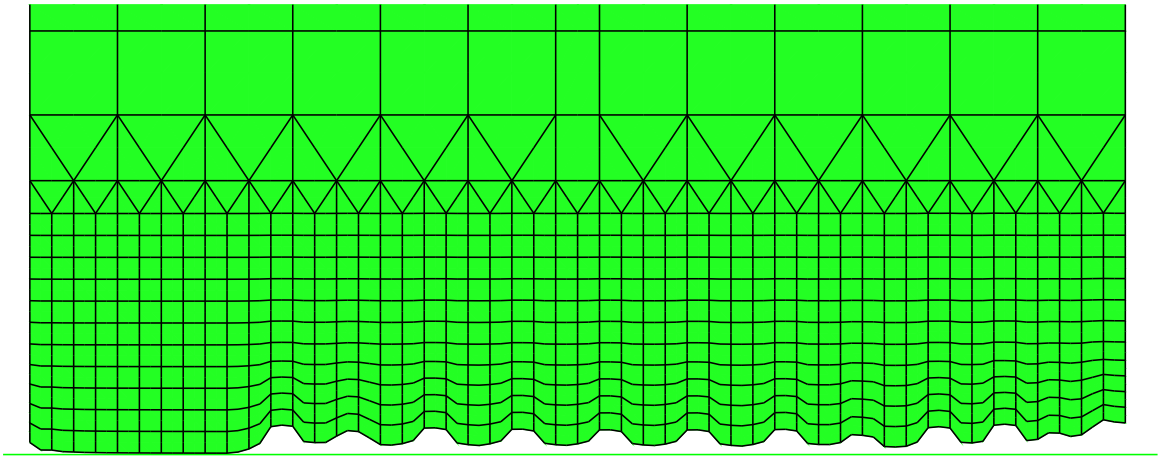


Figure B.13: Wear profile after 130m

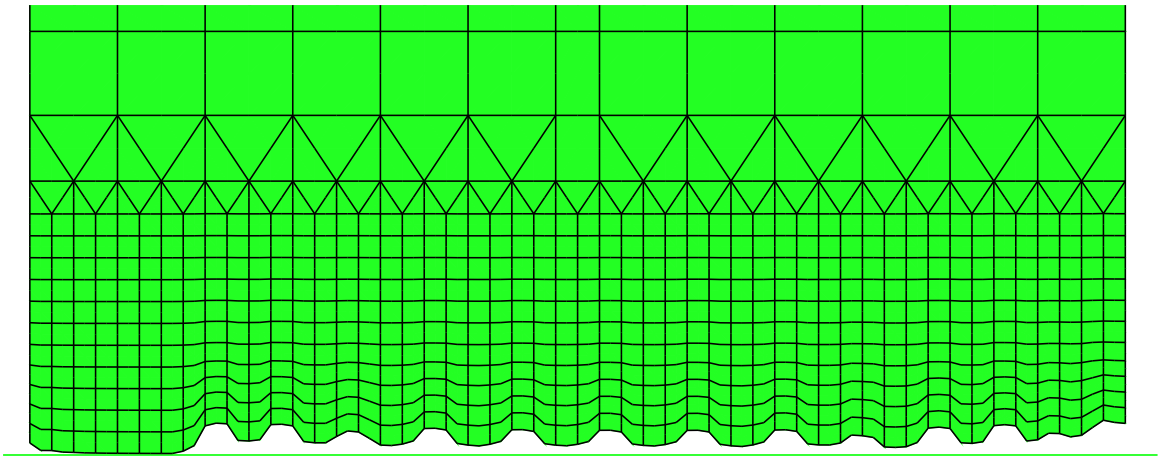


Figure B.14: Wear profile after 140m

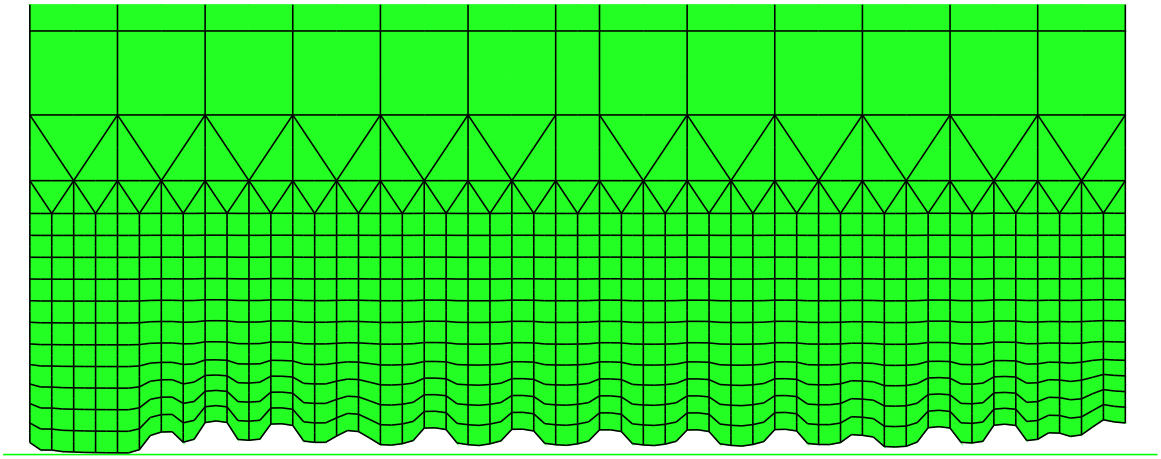


Figure B.15: Wear profile after 150m

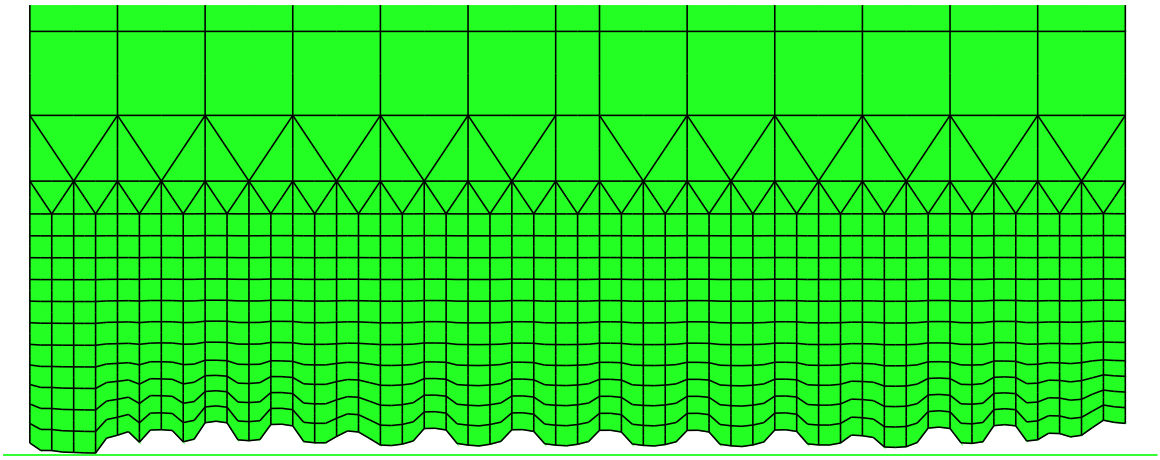


Figure B.16: Wear profile after 160m

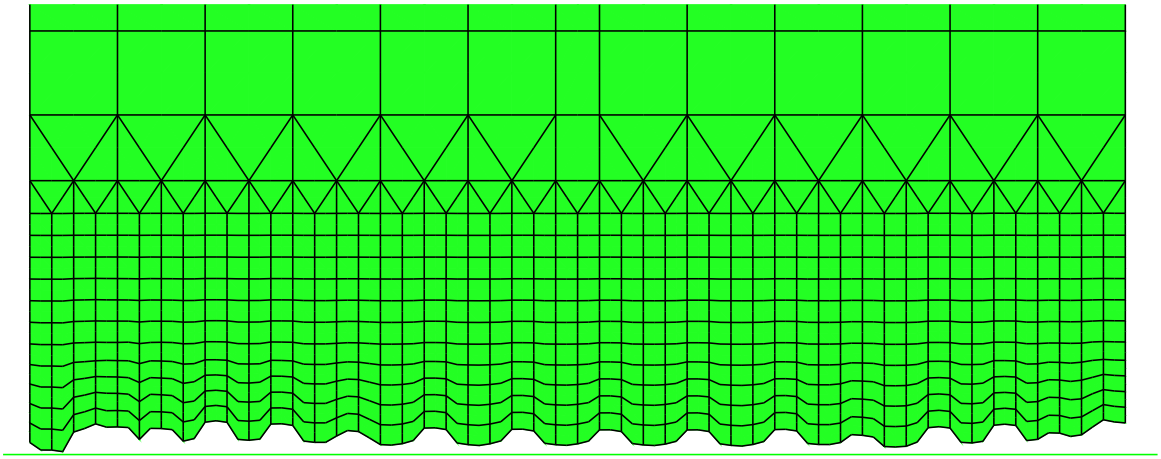


Figure B.17: Wear profile after 170m

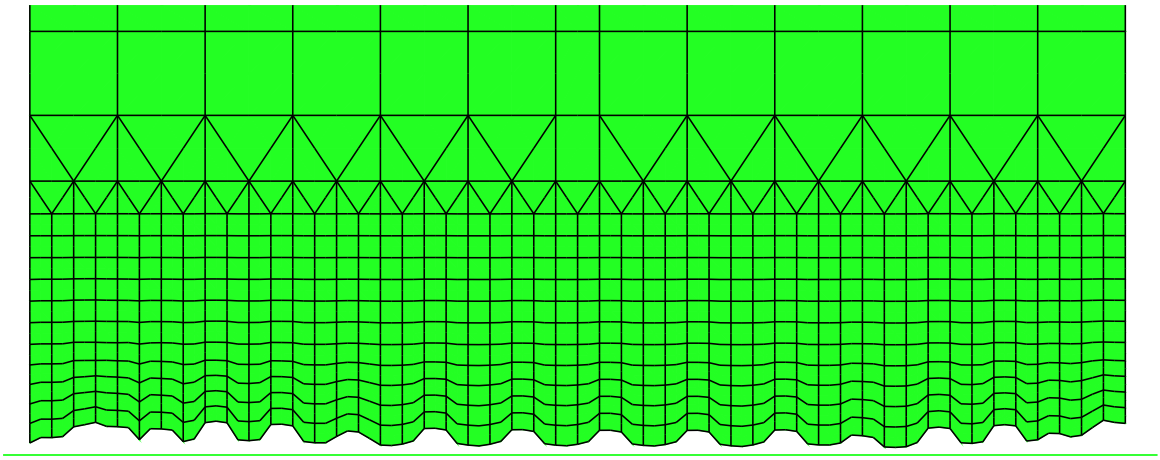


Figure B.18: Wear profile after 180m

Appendix C. Wear Progression Images for the Second Test of the Incremental Method

This appendix contains images of the wear profile after every 10 steps of the script for the second case described in chapter IV. Only the bottom portion of the pin is shown to conserve space. The progression of the high pressure spot from the rear of the pin to the front is clear. Each image is after another 0.1m of travel. The pin is traveling to the left.

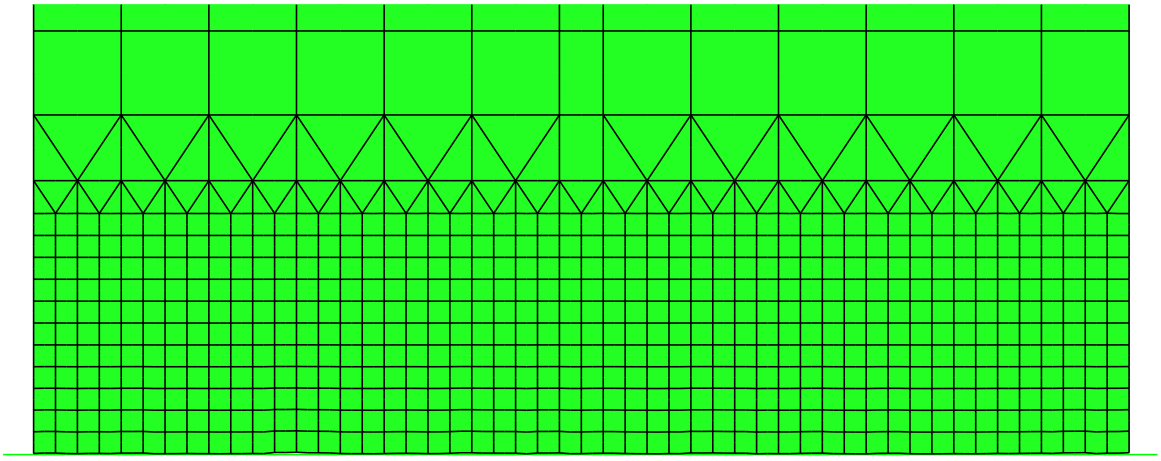


Figure C.1: Wear profile after 0.1m

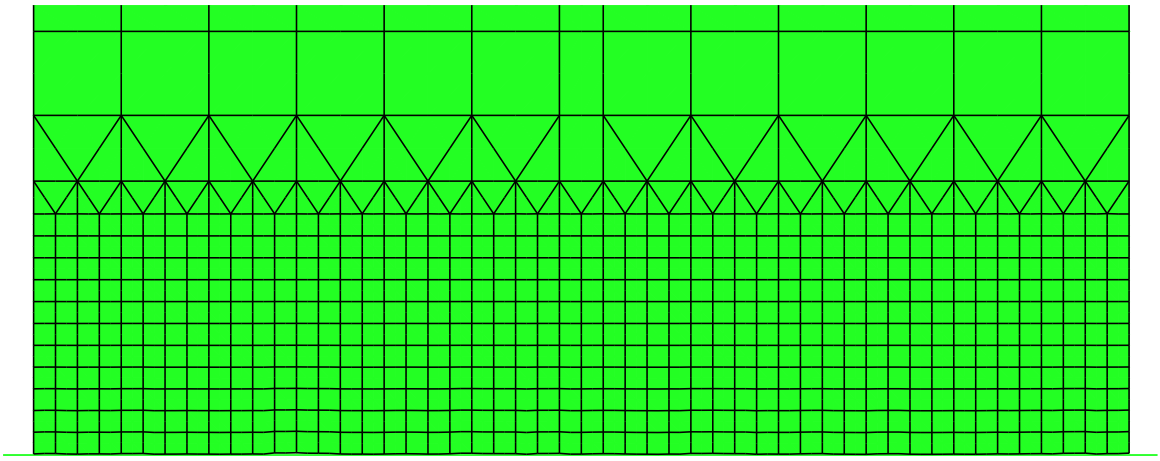


Figure C.2: Wear profile after 0.2m

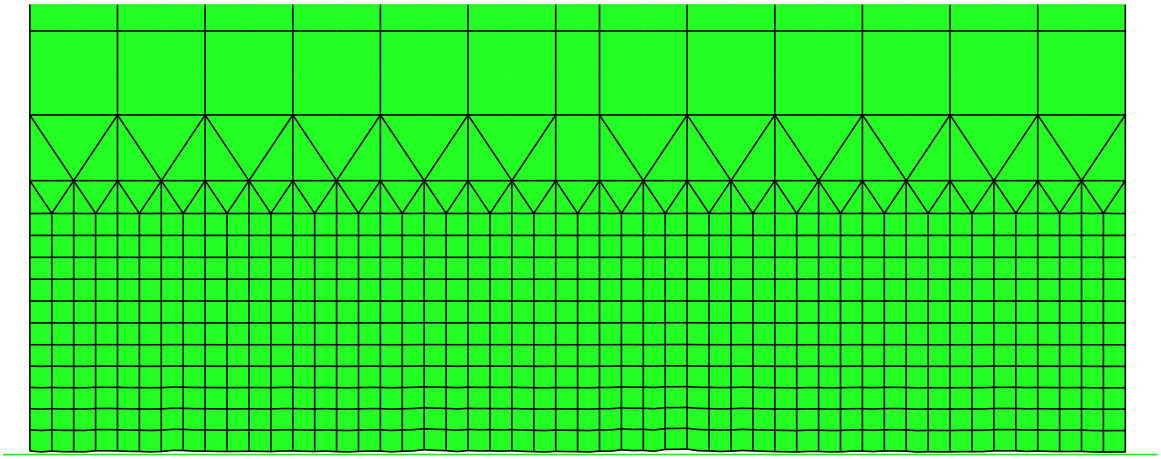


Figure C.3: Wear profile after 0.3m

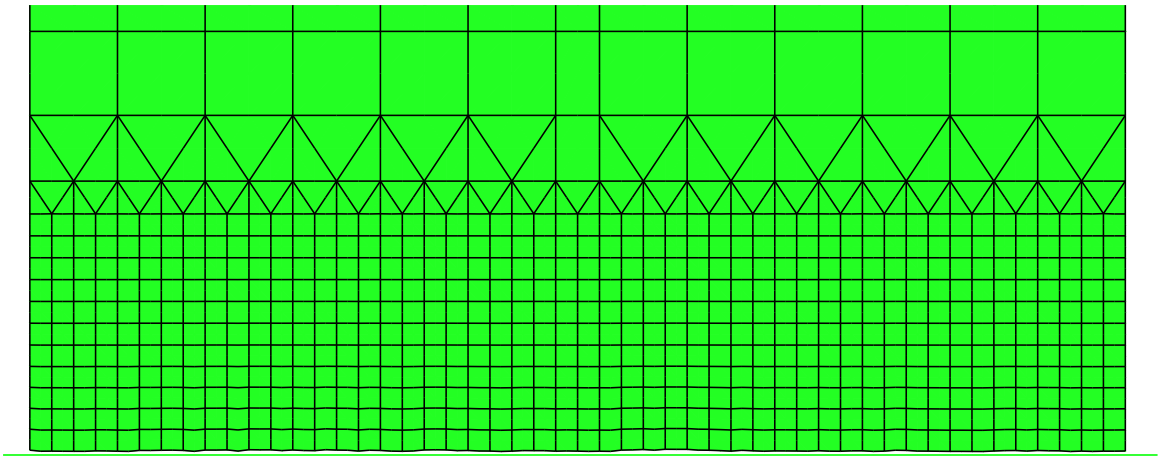


Figure C.4: Wear profile after 0.4m

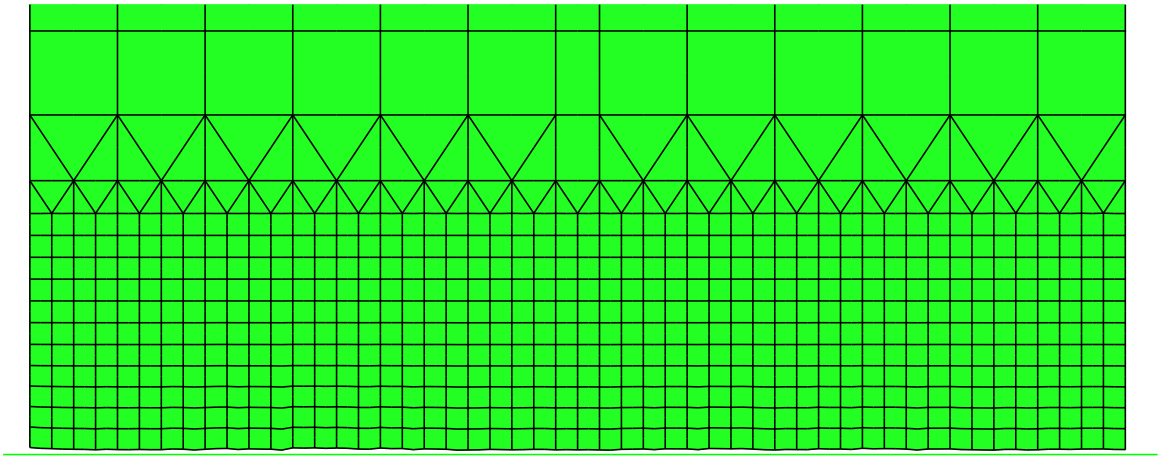


Figure C.5: Wear profile after 0.5m

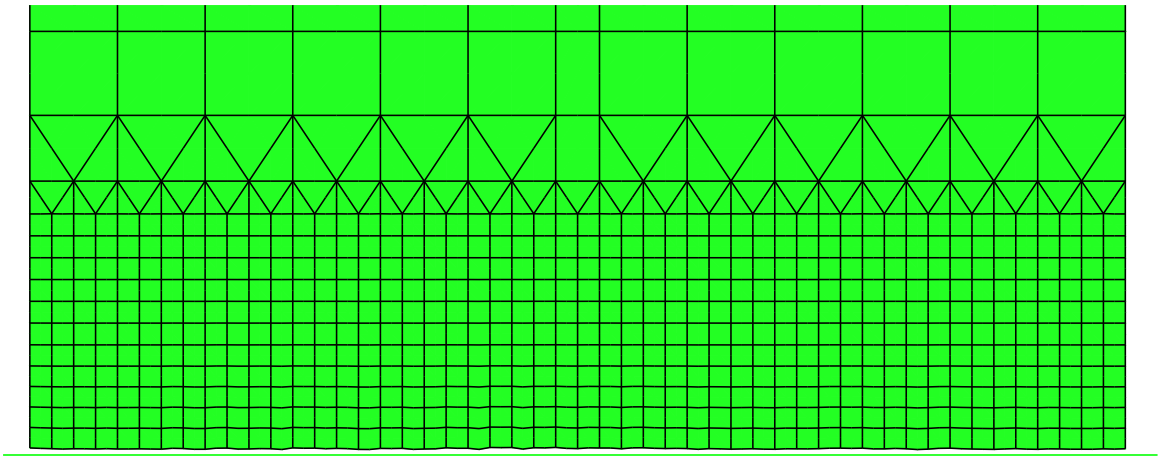


Figure C.6: Wear profile after 0.6m

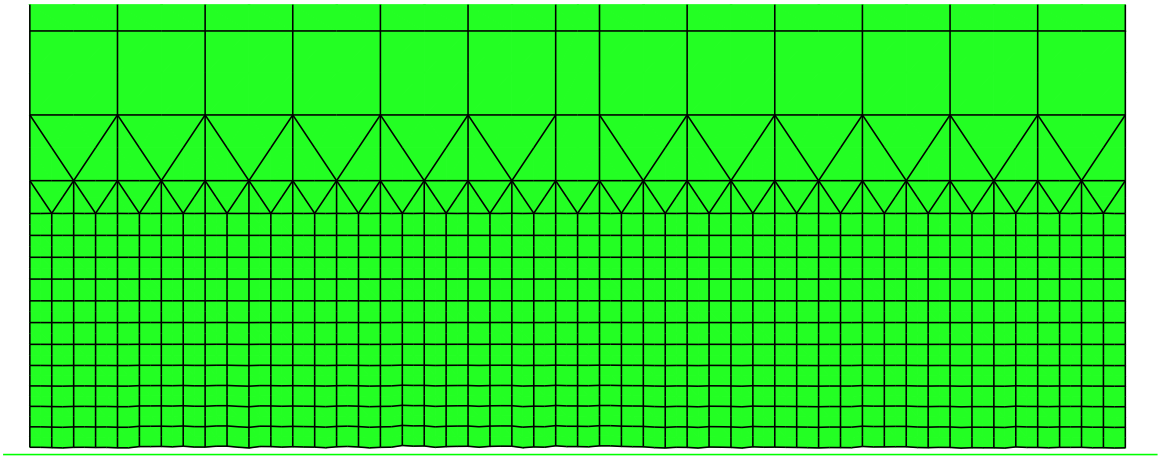


Figure C.7: Wear profile after 0.7m

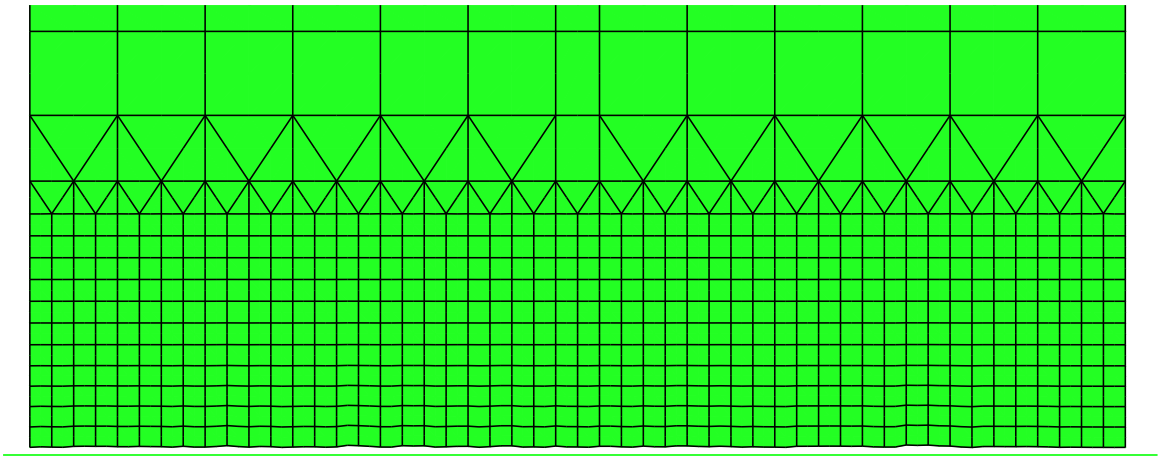


Figure C.8: Wear profile after 0.8m

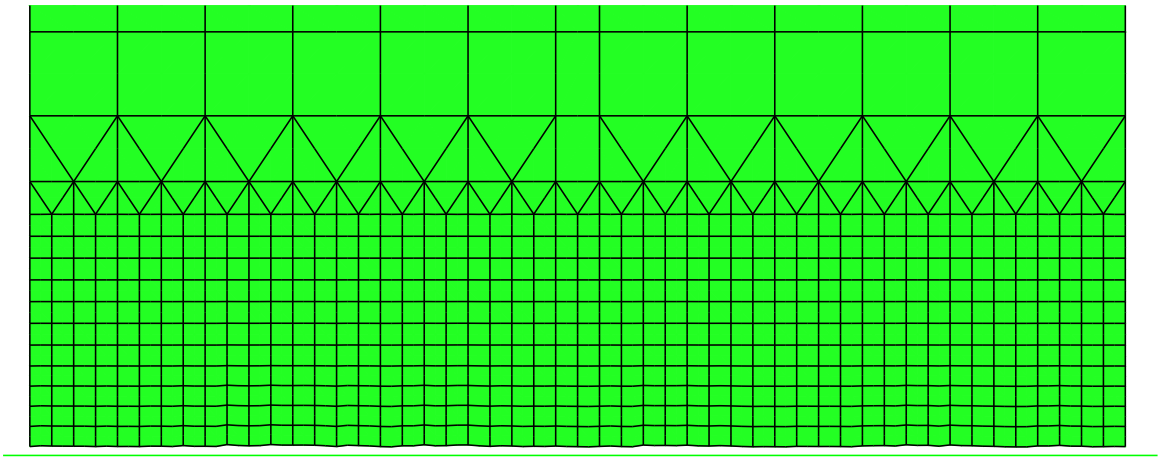


Figure C.9: Wear profile after 0.9m

Bibliography

1. *Holloman High Speed Test Track Design Manual*, January 2008.
2. Ayers G. "DADS File for Nike Sled 80X-A1." Unpublished manuscript, 2007.
3. Bayer R. G. *Mechanical Wear Preditction and Prevention*. New York: Marcel Dekker, 1994.
4. Benabdallah H. and Olender D. "Finite Element Simulation of the Wear of Polyoxymethylene in Pin-On-Disk Configuration," *Wear 261*, 1213–1224 (2006).
5. Cameron G. J. *An Evaluation of High Velocity Wear*. MS thesis, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 2006. AFIT/GAE/ENY/07-M06.
6. Cinnamon J. D. *Analysis and Simulation of Hypervelocity Gouging Impacts AFIT/DS/ENY/06-01*. PhD dissertation, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, June 2006.
7. Cook R. D., Malkus D. S., Plesha M. E., and Witt R. J. *Concepts and Applications of Finite Element Analysis* (4th Edition). Hoboken NJ: John Wiley and Sons, Inc., 2002.
8. Dassault Systemes. *Abaqus/Explicit: Advanced Topics*, 2006. Course Notes.
9. Dassault Systemes. *Abaqus Analysis Users Manual* (6.7-1 Edition), 2007.
10. Dassault Systemes. *Abaqus Example Problems Manual* (6.7-1 Edition), 2007.
11. Hegadekatte V., Huber N., and Kraft O. "Finite Element Based Simulation of Dry Sliding Wear," *Modelling Simul. Mater. Sci Eng.*, *13*:57–75 (2005).
12. Hooputra H., Gese H., Dell H., and Werner H. "A Comprehensive Failure Model for Crashworthiness Simulation of Aluminum Extrusions," *IJCrash*, *9*(5):449–463 (2004).
13. Hooser M. "Rail dimensions." Unpublished manuscript, 2007.
14. Hutchings I. M. *Tribology: Frition and Wear of Engineering Materials*. Boca Ratan: CRC Press, 1992.
15. Kurzydlowski K. J. and Ralph B. *The Quantitative Description of the Microstructure of Materials*. Boca Ratan: CRC Press, 1995.
16. Laird D. J. *The Investigation of Hypervelocity Gouging AFIT/DS/ENY/02-01*. PhD dissertation, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 2002.
17. Lim S. C. and Ashby M. F. "Wear-Mechanism Maps," *Acta metall.*, *35*(1):1–24 (1987).

18. Molinari J., Ortiz M., Radovitzky R., and Repetto E. "Finite-Element Modeling of Dry Sliding Wear In Metals," *Engineering Comptations*, *18*(3/4):592–609 (2001).

19. Montgomery R. S. "Friction and Wear at High Sliding Speeds," *Wear*, *36*:275–298 (1976).

20. Podra P. and Andersson S. "Simulating Sliding Wear with Finite Element Method," *Tribology International 32*, 71–81 (1999).

21. Prasad S. N., Chiu P. K., and Dasgupta S. "Compression and Sliding of an Elastic Rectangle Fixed Rigidly at the Base," *Int. J. Engng Sci.*, *14*:617–629 (1976).

22. Pugh B. *Friction and Wear*. London: Newns-Butterworths, 1973.

23. Sarkar A. D. *Wear of Metals*. New York: Pergamon Press, 1976.

24. Szmerekovsky A. G. *The Physical Understanding of the use of Coatings to Mitigate Hypervelocity Gouging Considering Real Test Sled Dimensions AFIT/DS/ENY/04-06*. PhD dissertation, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 2004.

25. Thompson J. M. and Thompson M. K. "A Proposal for the Calculation of Wear,"

26. Unknown . "46th Test Wing Fact Sheet." 846th Test Squadron.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE (DD–MM–YYYY)	2. REPORT TYPE	3. DATES COVERED *(From — To)*
14–03–2008	Master's Thesis	Mar 2007 — Mar 2008

4. TITLE AND SUBTITLE

Finite Element Simulation Methods for Dry Sliding Wear

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Aaron J. Chmiel

5d. PROJECT NUMBER
Jon# 169

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GAE/ENY/08-M03

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Office of Scientific Research
Attn: Dr. John Schmisseur
4015 Wilson Blvd, Rm 713
Arlington, VA 22203-1954

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR/NM

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Holloman High Speed Test Track is a rocket sled track for testing at hypersonic velocities. However, there are customers that desire to test at even greater velocities. In order to achieve higher velocities there are several phenomena that must be overcome. One important phenomenon is wear of the shoe that holds the sled on the rail. This research is a look at the feasibility of using finite element analysis to predict the wear of the shoe during a test run down the track. Two methods are investigated, one a macro-scale, incremental method utilizing traditional wear equations from Archard, the other a micro-scale, material property method that used a failure criteria to determine the amount of wear. These methods are implemented at low speeds to allow for comparison to results from the literature. While the incremental method was found to provide accurate results, there are many numerical problems associated with it. The material property method was found to be feasible, but more research is needed to validate and calibrate the process.

15. SUBJECT TERMS

Wear, Test Facilities, Finite Element Analysis, Failure

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Anthony Palazotto
U	U	U	UU	123	19b. TELEPHONE NUMBER (Include Area Code) (937) 255-3636, ext 4599 e-mail: anthony.palazotto@afit.edu